1972

Convergence acceleration and error analysis of the discrete ordinates algorithm in plane geometry.

Atkinson, Gerald Lloyd

University of Michigan

CONVERGENCE ACCELERATION AND ERROR ANALYSIS OF THE DISCRETE ORDINATES ALGORITHM IN PLANE GEOMETRY

by

Gerald Lloyd Atkinson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Nuclear Engineering)
in The University of Michigan
1972

Doctoral Committee:

Professor William Kerr, Chairman
Assistant Professor Melvyn Ciment
Assistant Professor James J. Duderstadt
Associate Professor Glenn F. Knoll
Associate Professor Fred C. Shure

ACKNOWLEDGMENTS

The author gratefully acknowledges the support that Professor William Kerr has provided while acting as his dissertation committee chairman. His efforts on the author's behalf to obtain permission from the U.S. Navy to return to academic pursuits and to remain until completed is deeply appreciated. His powerful physical intuition, ability to pose the appropriate question, and thoughtfulness in accommodating the needs of his students will long be remembered.

The assistance and guidance of Professor James J. Duderstadt in the mathematical details of this work were important ingredients in its completion. The author is also indebted to Kamal A. El-Sheikh for the many fruitful discussions which provided continuous motivation for completion of this work.

The author is deeply grateful to the government of the United States of America for the financial support through the U.S. Navy without which this work and the author's continued education would have been impossible.

Support for computing was provided by The University of Michigan Computing Center.

The most important support, which can never be fully rewarded, was received from the author's wife, Arvie, whose

selfless devotion to the additional responsibilities imposed on her and sacrifices suffered by her, allowed the author's single-minded pursuit of the goal.

TABLE OF CONTENTS

Page

Page

APPENDICES

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$\psi(x,\mu)$	Angular flux solution of analytic transport equation in plane geometry
$\psi_j(x)$	Solution of S_N equations
$\underline{\psi}^*$	Exact vector solution of spatially discretized S_N algorithm
$\psi_{k,j}^{(i)}$	Component of i^{th} iterate approximation to $\underline{\psi}^*$
$\underline{\psi}^{(i)}$	Vector representation of solution set $\{\psi_{k,j}^{(i)}\}$
$\psi_k^{(i)}$	k^{th} component of vector $\underline{\psi}^{(i)}$
L	Matrix operator for fully discretized system
D-E	Matrix containing the effects of streaming and collision
S	Scattering matrix
$\\| \quad \\|$	Norm of the matrix or vector within
$q(x,\mu)$	Inhomogeneous source term
x	Spatial variable
μ	Direction cosine of angle between particle motion and the X-axis
w_j	j^{th} weight in the quadrature set $\{w_j,\mu_j\}$
σ^t	Macroscopic total cross section
σ^s	Macroscopic scattering cross section
Δ	Spatial increment of the mesh
R	Number of spatial increments in the slab width
N	Number of angular ordinates in the set $\{\mu_j\}$
$\underline{e}^{(i)}$	Iteration error in the i^{th} iterate approximation
d_j	Coefficient in the unmodified S_N algorithm

e_j	Coefficient in the unmodified S_N algorithm
$\underline{\delta}^{(i)}$	Difference between successive iterate solutions
α	Acceleration parameter, positive real constant
ε	Arbitrarily small positive real constant
β	Positive real constant
γ	Positive real constant
$\phi_j(x)$	Transform domain solution of angular discretized S_N equations
$\underline{\phi}^*$	Exact vector solution of spatially discretized transform domain S_N algorithm
$\phi_{k,j}^{(i)}$	Component of i^{th} iterate approximation to $\underline{\phi}^*$
$\underline{\phi}^{(i)}$	Vector representation of solution set $\{\phi_{k,j}^{(i)}\}$
$\phi_k^{(i)}$	k^{th} component of vector $\underline{\phi}^{(i)}$
e_j	Coefficient in transform domain algorithm
d_j	Coefficient in transform domain algorithm
$\mathcal{D}-\mathcal{E}$	Transform domain matrix with same form as D-E but with matrix elements containing the acceleration parameter α
$\tau_j(x)$	Discretization error
ρ	Condition number of iteration matrix (in Chapter V)
ν	Condition number of matrix D-E
δE	Uncertainty in the matrix E
δD	Uncertainty in the matrix D
$\underline{\delta q}$	Uncertainty in the vector $\underline{q}$
$\underline{\delta\psi}^{(i)}$	Uncertainty in the iterate solution $\underline{\psi}^{(i)}$

CHAPTER I

INTRODUCTION TO DISCRETE ORDINATES METHODS IN TRANSPORT THEORY

Transport Theory

The behavior of a nuclear reactor and the effectiveness of its shield are governed by the distribution in space, velocity, and time of the neutrons and photons in the system. The transport equation is a statement of conservation for these neutral particles, the solution of which determines their distribution in phase space. Derivation of the transport equation is presented in most transport theory texts.[1-3]

A few special systems have exact solutions,[4,5] but for more practical systems, numerical solutions of an approximate transport equation are sought. A few of the more useful transport approximations are spherical harmonics[6] Monte Carlo,[7] discrete ordinates,[8] and moments methods.[9]

The spherical harmonics method treats the anisotropy of the particle distribution and cross sections to various degrees of approximation. Its application to plane, spherical, and cylindrical geometries has been useful but for more complex geometries, the spherical harmonics method is so complicated that other methods are used.

The moments method, used in shielding studies, calculates particle transport rigorously but the method is limited to infinite homogeneous media.

The discrete ordinates and Monte Carlo methods are the most nearly rigorous for multidimensional problems but the former is hampered by ray-effects[10] and is currently limited to two-dimensional problems. The Monte Carlo method is the only useful technique for rigorous solution of three-dimensional problems.

Development of Discrete Ordinate Method

The use of discrete ordinates was first suggested by Wick in 1943[11] and developed for radiation transport in stellar atmospheres by Chandrasekhar.[12,13] Carlson introduced the angular "segmented" S_N approximation for radiation transport calculations at Los Alamos in late 1952 and early 1953.[14] A complete history of S_N development is given by Carlson[15] and a current bibliography is available.[16] Chernick[17] gives a detailed historical review of reactor physics calculation methods from the Manhattan Project of the 1940's to the present and describes the place of S_N computer codes in reactor criticality and shielding applications.

Early S_N methods have been refined to the present "diamond difference" scheme, described by Carlson,[8] which is incorporated in most S_N codes. New approximations, such as MS_N,[18] have been derived for special problems.

In 1967 Shreiner[19] attempted to derive an S_N difference scheme, called VS_N, from a variational principle but produced no improvement over existing methods. More recently, effort has been expended to derive discrete ordinate approximations from space-angle synthesis techniques.[20] Kaplan[21] derives an S_2 approximation in X,Y geometry in this manner by using step functions in direction as trial functions to eliminate ray effects. Natelson[22] derives discrete ordinate approximations from a first-order variational principle in X,Y geometry. For some unknown reason, this approach has not led to improved difference schemes over the standard diamond difference S_N method.

Applications of S_N Method

Transport theory, as opposed to one of the simpler approximation techniques such as diffusion theory, must be used when either very precise solutions are required or flux variations are large. As a rule of thumb, transport theory is used when the logarithmic gradient of the particle flow density is of the order of or larger than an inverse mean free path over significant portions of the system. Early application of S_N by Lee[23] to the one-speed problem revealed that the S_8 approximation gave critical radii within 0.3% of the values obtained by the exact solution of the neutron transport equation. Problems exist, however, for which the one-speed approximation is not sufficient. Mills[24] applying S_N to small fast critical assemblies such as Godiva, Jezebel, and Topsy, revealed that 24-group, S_8

calculations with a P_2 approximation to the scattering anisotropy should be sufficient to provide an accurate treatment of neutron transport in fast metal assemblies.

More recently, the demand for tighter safety standards has awakened renewed interest in application of transport theory S_N approximations. Protsik[25] compared results of multigroup diffusion theory and S_N calculations and found that significant error in loss-of-coolant reactivity calculations occurs using diffusion theory. He found that diffusion theory over-predicts the neutron leakage and under-predicts the multiplication.

The S_N approximation has also been used as a standard, for problems without an exact solution, by which the merits of other techniques are judged.[26-28] S_N solutions of the adjoint equation have been used in importance sampling techniques for Monte Carlo methods.[29]

Multigroup S_N Equations

The steady-state multigroup transport equations can be written[30]

$$L\psi(\underline{r},\hat{\Omega}) = S\psi(\underline{r},\hat{\Omega}) + q(\underline{r},\hat{\Omega}) \tag{1}$$

where the vectors ψ and q contain the unknown distributions and sources in each energy group, respectively. The operators L and S take the following forms

$$\left(L\psi(\underline{r},\hat{\Omega})\right)_g = \hat{\Omega}\cdot\nabla\psi_g(\underline{r},\hat{\Omega}) + \sigma(\underline{r})\psi_g(\underline{r},\hat{\Omega}) \tag{2}$$

$$\left(S\psi(\underline{r},\hat{\Omega})\right)_g = \sum_{g'} \int_{\hat{\Omega}'} d\hat{\Omega}'\, \mathfrak{f}(\underline{r};\hat{\Omega}',g' \rightarrow \hat{\Omega},g)\psi_{g'}(\underline{r},\hat{\Omega}') \tag{3}$$

where the subscript g denotes the g^{th} energy group.

The S_N approximation discretizes the above equations at N discrete angular ordinates, $\hat{\Omega}_n$, to produce the set

$$L\psi(\underline{r},\hat{\Omega}_n) = S\psi(\underline{r},\hat{\Omega}_n) + q(\underline{r},\hat{\Omega}_n) \tag{4}$$

in which

$$\left(L\psi(\underline{r},\hat{\Omega}_n)\right)_g = \hat{\Omega}_n\cdot\nabla\psi_g(\underline{r},\hat{\Omega}_n) + \sigma(\underline{r})\psi_g(\underline{r},\hat{\Omega}_n) \tag{5}$$

and

$$\left(S\psi(\underline{r},\hat{\Omega}_n)\right)_g = \sum_{g'}\sum_{m=1}^{N}\omega_m\delta(\underline{r};\hat{\Omega}_m,g' \rightarrow \hat{\Omega}_n,g)\psi_{g'}(\underline{r},\hat{\Omega}_m) \tag{6}$$

where ω_m are weights associated with the angular quadrature set $\{\hat{\Omega}_m\}$ and the sum over m approximates the integration over $\hat{\Omega}'$ in (3). Equations (4), (5), and (6) describe the S_N approximation to the transport equation.

Finite-difference techniques using the diamond difference approximation[8] have been applied which approximate equation (4) to second order in all independent variables in Cartesian geometries and to order $\Delta r^2/r^2$ in one-dimensional spherical geometry.[31] Applying any method of spatial discretization, the S_N equations can be written in the matrix form

$$L\underline{\psi} = S\underline{\psi} + \underline{q} \tag{7}$$

where the vectors $\underline{\psi}$ and $\underline{q}$ now contain the unknown flux and inhomogeneous source, respectively, in each group and at all mesh points, space and angle. The matrices L and S are approximations to the operators L and S respectively.

Outer Iteration

The scattering matrix S is split as follows

$$S = S_d + S_s + S_u \tag{8}$$

where S_d, S_s, and S_u represent down scattering, self or in-group scattering, and up-scattering respectively. The iterative technique

$$(L - S_d - S_s)\underline{\psi}^{j+1} = S_u\underline{\psi}^j + \underline{q} \tag{9}$$

represents the "outer" iteration where j is the iteration index. This matrix equation is never explicitly solved in transport codes. The matrix $(L-S_d-S_s)$ is dense and of such large dimension that it is not easily invertible on present day computers. Consequently another splitting is made so that an inner iteration can be conducted within each energy group.

Inner Iteration

The self or in-group scattering matrix is separated from the left hand side of (9) to operate on the k^{th} inner iterate group flux,

$$L_g\psi_g^{j+1,k+1} = S_{sg}\psi_g^{j+1,k} + s_g^{j+1} \tag{10}$$

where the source to the g^{th} group is

$$s_g^{j+1} = (S_d\underline{\psi}^{j+1})_g + (S_u\underline{\psi}^j)_g + q_g \ . \tag{11}$$

In this inner iteration scheme, the first term on the right-hand-side of (11) can be evaluated even though $\underline{\psi}^{j+1}$ is not

known for groups g and below, since elements of S_d corresponding to these groups are zero. The matrices L_g and S_{sg} in (10) are restrictions of L and S_s to the g^{th} group. The vector ψ_g^{j+1} contains components of $\underline{\psi}^{j+1}$ corresponding to the g^{th} group. The matrix L_g is triangular and thus easily invertible for typical difference schemes, so $\underline{\psi}_g^{j+1,k+1}$ can be obtained rapidly from $\underline{\psi}_g^{j+1,k}$.[30]

A converged inner iterate solution in each energy group completes one outer iteration. Consequently the outer iterate equation (9) is never solved but its solution is approximated by completing the inner iteration for each group a number of times, each with a source, $\underline{q}$, calculated from previous outer iteration fluxes. For media which do not reproduce neutrons, only one outer iteration is required but for fissioning media, many outer iterations are required.

The problems treated here employ the inner iteration of equation (10) for plane geometry. The specific equations will be developed later.

Convergence of S_N Approximation

Early versions of the discrete ordinates equations were used and results compared to those of other methods without regard to their formal convergence properties. Numerical experiments with problems representing physical systems showed that the early (circa late 1950's) time-dependent schemes were stable.[32] Not until 1960 did proof of convergence of the steady-state S_N approximation appear when Keller[33] proved mean-square convergence of the S_N

approximation to the analytic solution in plane geometry, one-speed, with isotropic scattering for values of $0 \leq C < 1$ where C is the mean number of secondary neutrons per collision. At the same time, Wendroff[34] proved pointwise convergence for the same problem for $C < 1$ and a weighted mean-square convergence for $C = 1$. Keller[35] immediately thereafter proved pointwise convergence for the problem for values of C greater than unity.

Not until 1968 did convergence proofs appear for multidimensional geometries. Madsen[36-38] proved pointwise convergence of the S_N approximation to the time-independent one-speed angular segmented transport solution in x,y geometry with vacuum and periodic boundary conditions. In addition he proved that the spatially differenced approximations using the central difference, first-order difference, and diamond difference schemes converge pointwise to the angular S_N approximation solution in x,y geometry. Madsen[39] has shown that the S_N approximation in x,y,z geometry converges pointwise to the exact solution under certain conditions.

Acceleration of Convergence

Current work is being directed toward improved techniques to accelerate convergence of the S_N algorithm. Clifford[40] gives a lucid account of S_N running times for shielding problems using various generation digital computers. With present generation IBM/360 series machines, very large two-dimensional problems require 10 to 20 hours

running time. For deep penetration problems in media with σ^s/σ^t near unity, Reed[30] states that non-accelerated S_N methods converge so slowly that they are practically useless.

Several methods have been developed to accelerate inner iteration convergence of the S_N algorithm. Carlson and Bell[41] proposed a scale factor technique which is used in present standard codes like ANISN. It uses a system-wide neutron conservation principle and iterates until a "false" source is arbitrarily small.[8] Engle[42] found that the scale factor technique accelerates convergence rapidly for absorbing media and near source regions for scattering media but is slow to converge at points many mean free paths away from the source in predominantly scattering media. For the latter problems, the scale factor rapidly approaches unity (even when the group flux solution is far from converged) and thereafter does little to accelerate convergence. Consequently, he proposed a separate scale factor for each space interval which exhibited some success in a one-dimensional 26-group problem. Clancy[43] proposed an outer iteration scaling technique which, when used with inner iteration scaling, accelerates convergence for problems characterized by the presence of significant upscattering.

The acceleration technique often used in one-dimensional problems is that of Chebychev, adapted to transport

approximations by Hageman.[44] It is not as effective in two-dimensional geometries.

The Synthetic Method, first proposed by Kopp[45] and applied by Gelbard[46] has been quite useful in accelerating convergence of the S_N algorithm in one and two-dimensional problems. This method uses the solution of a diffusion theory approximation to accelerate convergence of the iterative S_N algorithm.

Another effective method of accelerating convergence is called coarse mesh rebalancing. An early version was suggested by Wachspress[47] for diffusion theory codes. The method has been applied to a transport code, TWOTRAN.[48] A variational rebalancing scheme was devised by Nakamura[49] and applied to the one-dimensional S_N algorithm.

The most effective acceleration methods proposed to date are the synthetic and coarse mesh rebalance methods. When successful, they lead to tremendous reductions in computing time. Unfortunately, Reed[30] discloses, there are problems for which the use of these acceleration techniques lead to an unstable algorithm. This failure occurs on those problems where it is most necessary to accelerate convergence, namely problems with an optically thick region with scattering ratio near unity. He displays model problems for which the unaccelerated S_2 algorithm requires more than 2600 iterations to converge and the above two methods fail to converge. He also generates parameters which force both methods to converge.

A typical problem for which acceleration techniques are required is neutron propagation through large thicknesses of iron which has a huge scattering resonance at about 25 kev. Devillers[50] compares S_N codes running times (NIOBE, 20 minutes; ANISN, 8 minutes) to a Monte Carlo code (POKER, 13 minutes) for a one-dimensional multigroup problem. Deep penetration problems in iron have also been solved by S_N techniques to study the "window" effect for such materials.[51]

Current research is directed toward developing more accurate transport results. Increased emphasis on reactor safety is one factor responsible for this trend. But it is not sufficient to strive for accuracy alone; efficiency or cost must also be considered. Consequently, acceleration techniques play a prominent role in present computer codes. Any technique which contributes to improved efficiency will be incorporated to reduce the expensive computer computational costs.

Proposed Investigation

This investigation formulates the angular flux form of the S_N equations in plane geometry within a mathematical frame which allows an insight into the mechanics of the convergence process. This mathematical formulation leads to sufficiency conditions under which iterative convergence to an exact discretized solution is assured. These conditions include a domain in which some elements of the iteration matrix are allowed to be negative.

The infinity-norm of the iteration matrix is calculated for a problem with a homogeneous isotropically scattering medium. This allows the calculation of an improved convergence criterion. Reed[52] states that the convergence criterion

$$\max_{i} \left| 1 - \frac{\phi_i^{(k-1)}}{\phi_i^{(k)}} \right| \leq 0.0001$$

where $\phi_i^{(k)}$ is the i^{th} component of the k^{th} scalar flux iterate is not, strictly speaking, a measure of the error in the final iterate, since slowly converging methods may meet this test without being close to the exact solution. It is, however, the convergence criterion actually used in most transport codes. This investigation proposes an improved convergence criterion which, along with the demonstrated convergence properties of the S_N algorithm, guarantees that the fractional iterative error, defined in Chapter II, is arbitrarily small. This is exhibited in Chapter II.

A spatial transform method which accelerates inner iteration convergence for certain S_N problems is developed in Chapter III. The method is applicable to problems for optically thick media in which the scattering ratio is near unity. The transform renders the angular discretized S_N equations invariant and places an acceleration parameter in each non-zero matrix element of the iteration matrix such that its norm is reduced. This results in accelerated convergence of the S_N algorithm.

The reciprocal of the spatial transform is applied to an S_N problem to reduce the discretization error. This variation of the transform method is applied to a predominantly absorbing medium problem in Chapter IV.

An upper bound is found for the round-off error in the S_N iterative algorithm describing an inner iteration of a multigroup problem. The expression for this bound provides insight into the reasons why iterative techniques often exhibit smaller roundoff error than competitive matrix inversion methods and how these errors propagate through the iterative process. This exposition is presented in Chapter V.

CHAPTER II

DISCRETE ORDINATES AS A METHOD OF SUCCESSIVE APPROXIMATIONS

To analyze the convergence properties of the S_N algorithm, it is necessary to provide some theoretical background and to show that the algorithm possesses, under certain restrictions, sufficient conditions for convergence.

Theoretical Background

The problem to be solved is

$$L\underline{\psi}^* = \underline{q} \tag{1}$$

where L is a finite dimensional square matrix, $\underline{\psi}^*$ is the solution vector, and $\underline{q}$ is the inhomogeneous vector. All vector and matrix elements are real. For the applications considered here, L is non-singular so equation (1) has a unique solution, $\underline{\psi}^*$, for each given $\underline{q}$.

Now assume that L can be split as follows,

$$L = D - E - S \, . \tag{2}$$

The choice of the splitting is arbitrary but the implication is that $(D-E)^{-1}$ is an approximation to L^{-1} but $(D-E)$ is much simpler to invert than L. For instance $(D-E)$ may be chosen such that it is sparse compared to the dense L.

After splitting L, equation (1) becomes

$$(D-E)\,[I-(D-E)^{-1}S]\underline{\psi}^* = \underline{q}$$

or

$$\underline{\psi}^* = [I-(D-E)^{-1}S]^{-1}(D-E)^{-1}\underline{q}$$

where

$$L^{-1} = [I-(D-E)^{-1}S]^{-1}(D-E)^{-1}.$$

Use will now be made of a convergence theorem from Isaacson,[53] which is repeated here. The notation has been modified for convenience to conform to that of the S_N algorithm used later.

Theorem 1

The geometric series $\sum_{m=0}^{\infty}[(D-E)^{-1}S]^m$ converges if and only if

$$||(D-E)^{-1}S|| < 1 . \tag{3}$$

If $(D-E)^{-1}S$ is convergent, then $[I-(D-E)^{-1}S]$ is non-singular and

$$[I-(D-E)^{-1}S]^{-1} = \sum_{m=0}^{\infty}[(D-E)^{-1}S]^m .$$

If the splitting of L is such that expression (3) is satisfied, the theorem can be invoked to produce

$$L^{-1} = \sum_{m=0}^{\infty}[(D-E)^{-1}S]^m(D-E)^{-1}$$

which is the Neumann series expansion of L^{-1} and

$$\underline{\psi}^* = \sum_{m=0}^{\infty} [(D-E)^{-1}S]^m (D-E)^{-1}\underline{q} \; . \tag{4}$$

A consequence of the series expression for L^{-1} is that

$$||L^{-1}|| \leq \frac{||(D-E)^{-1}||}{1-||(D-E)^{-1}S||} \; . \tag{5}$$

Observe that $||L^{-1}||$ is bounded provided that $||(D-E)^{-1}S|| < 1$ and that the bound becomes tighter as $||(D-E)^{-1}S||$ approaches zero.

Iterative Process

On the basis of Theorem 1, Rall[54] constructs an iterative process, called the method of successive approximations, which has

$$\underline{\psi}^{(i+1)} = (D-E)^{-1}S\underline{\psi}^{(i)} + (D-E)^{-1}\underline{q} \tag{6}$$

where i is an iteration index. This process gives a sequence $\{\underline{\psi}^{(i)}\}$ of successive approximations which converge to the exact solution $\underline{\psi}^*$ starting from any initial guess $\underline{\psi}^{(o)}$. Equation (6) implies that

$$\underline{\psi}^{(i+1)} = \sum_{m=0}^{i} [(D-E)^{-1}S]^m (D-E)^{-1}\underline{q} + [(D-E)^{-1}S]^{i+1}\underline{\psi}^{(o)}. \tag{7}$$

Observe that $\lim_{i \to \infty} \underline{\psi}^{(i+1)} = \underline{\psi}^*$ since the first term on the right hand side of (7) approaches $\underline{\psi}^*$ and the second term vanishes due to condition (3).

Error Estimation

Using equations (4) and (7), it follows that the absolute error, defined by

$$\underline{e}^{(i)} = \underline{\psi}^{*}-\underline{\psi}^{(i)} \tag{8}$$

is bounded by

$$||\underline{e}^{(i)}|| \leq \frac{||(D-E)^{-1}S||^{i}}{1-||(D-E)^{-1}S||} \cdot ||(D-E)^{-1}\underline{q}|| \tag{9}$$

for $\underline{\psi}^{(0)} = 0$.

This error expression is of little practical value except to illustrate the idea of efficiency. It is desirable to obtain approximations of a given accuracy with the fewest number of iterations. Obviously, more iterations are required to achieve a given accuracy as $||(D-E)^{-1}S||$ approaches unity. Indeed the error is unbounded for $||(D-E)^{-1}S|| = 1$. But this condition violates expression (3) which guarantees that the series converges to L^{-1}. If $||(D-E)^{-1}S|| \geq 1$, the convergence of the iterative process is not guaranteed.

Formulation of S_N Algorithm

The S_N equations will be derived for an energy group in a multigroup problem applying the following assumptions to the transport equation:

1) steady state
2) infinite plane geometry with one spatial dimension

3) homogeneous medium

4) isotropic scattering in the Laboratory Coordinate System

5) no reentrant current at the slab boundaries.

The problem may be described by

$$\mu \frac{\partial\psi(x,\mu)}{\partial x} + \sigma^{t}\psi(x,\mu) = \frac{\sigma^{s}}{2}\int_{-1}^{1}\psi(x,\mu')d\mu' + q(x,\mu) \qquad (10)$$

with boundary conditions

$$\psi(o,\mu) = 0 \qquad \text{for } \mu > 0 \qquad (11a)$$

$$\psi(L,\mu) = 0 \qquad \text{for } \mu < 0 \qquad (11b)$$

where $X = 0$ at the left face of the slab and $X = L$ at the right face of the slab. Equation (10) is the balance equation for the angular flux in a particular energy group. The source term $q(X,\mu)$ represents neutrons scattered from other energy groups and born within the energy group, either from fission or external sources.

The angular and spatial discretization of equation (10) follows the procedure described by Bell.[55] Alternate formulations by Mynatt,[56] who applies numerical difference methods to the analytic balance equation, and Carlson,[8] who immediately writes a discretized form of the balance equation with unknown coefficients and evaluates them by applying neutron conservation principles, produce the same difference equations in plane geometry. Equation (10) is discretized in direction cosines by approximating it at each

of the N values of μ_j in the quadrature set selected. This gives the N coupled set of equations

$$\mu_j \frac{d\psi_j(x)}{dx} + \sigma^t \psi_j(x) = \frac{\sigma^s}{2} \sum_{n=1}^{N} \omega_n \psi_n(x) + q_j(x) \tag{12}$$

$$\text{for } j = 1,2,\cdots, N \quad \text{where } \mu_{\frac{N}{2}+j} = -\mu_j \tag{13}$$

for $j = 1,2,\cdots, N/2$.

A spatial mesh is selected with R equi-spaced increments providing $R+1$ space points, including the boundaries, at which $\psi_j(x)$ is to be approximated. Each increment, Δ, is defined by

$$\Delta = X_{k+1} - X_k \quad . \tag{14a}$$

The x,μ mesh is described in Figure 1.

The quantity $\psi_j(x)$ in equation (12) is approximated at each spatial midpoint of the mesh

$$X_{k+1/2} = \frac{X_{k+1} + X_k}{2} \tag{14b}$$

by

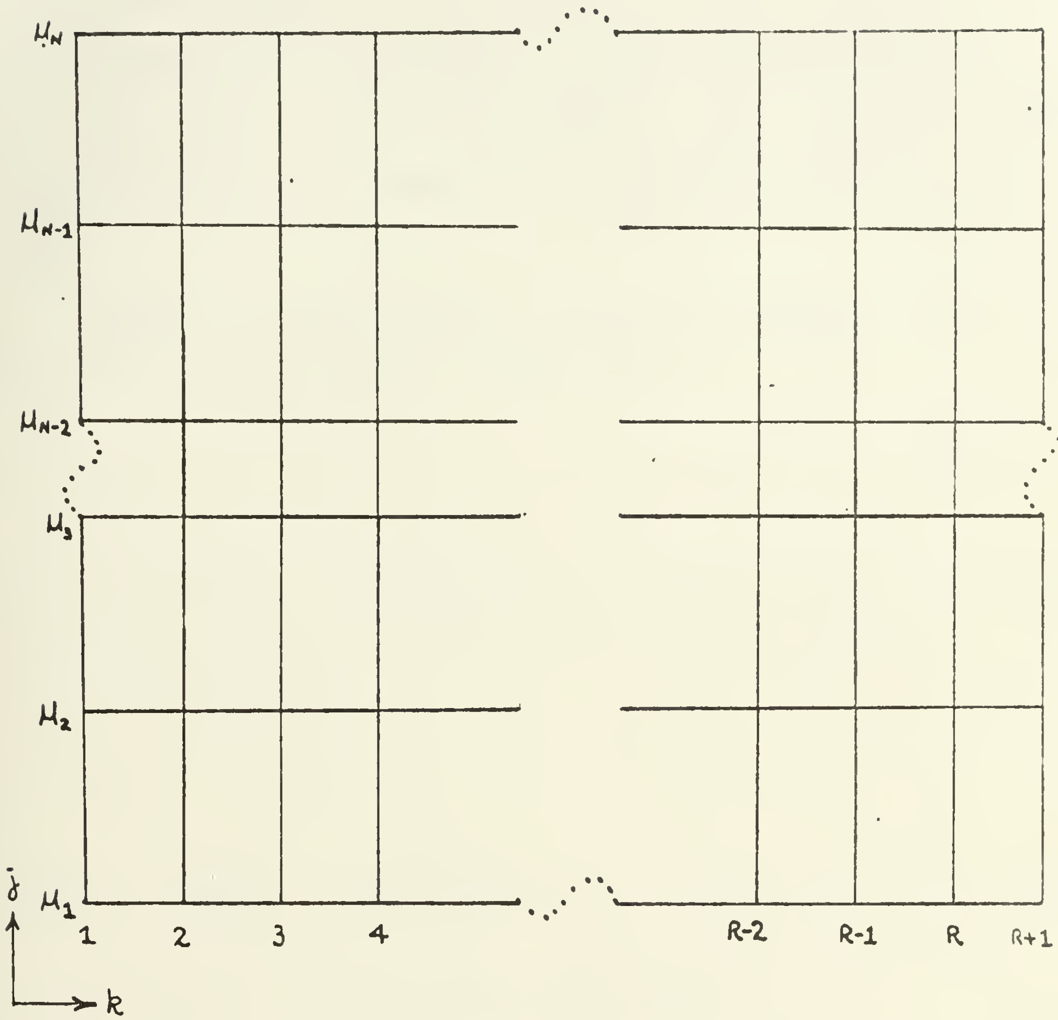
$$\psi_{k+1/2,j} = \frac{\psi_{k+1,j} + \psi_{k,j}}{2} \tag{14c}$$

and the derivative term is approximated by the central difference

$$\left.\frac{d\psi_j(x)}{dx}\right|_{X_{k+1/2}} = \frac{\psi_{k+1,j} - \psi_{k,j}}{\Delta} \quad . \tag{15}$$

FIGURE 2.1

SPACE-DIRECTION COSINE MESH



Direction set properties:

1) $\mu_{\frac{N}{2}+j} = -\mu_j, \; j = 1,2,\cdots,N/2 \; ; \; \mu_j > 0.$

2) $\sum_{n=1}^{N/2} \omega_n = 1, \; \omega_{\frac{N}{2}+j} = \omega_j \; , \quad j = 1,2,\cdots,\frac{N}{2} \; ; \; \omega_j > 0.$

This approximation of $\psi_{j(x)}$ by $\psi_{k,j}$ leads to a type of error called discretization error. Isaacson[53] shows that this approximation is accurate to order Δ^2, $O(\Delta^2)$. A desirable strategy is to choose Δ small enough that discretization error is reasonably small. This choice is based on a guess of how the solution is expected to vary across the slab.

The above substitutions applied to equation (12) produce the following set of equations.

For $\mu_j > 0$

$$d_j\psi_{k+1,j} - e_j\psi_{k,j} = \frac{\sigma^s}{4} \sum_{n=1}^{N} w_n(\psi_{k+1,n} + \psi_{k,n}) + q_{k+1/2,j} \quad (16a)$$

$k = 1,2,\cdots, R.$

For $\mu_j < 0$

$$d_j\psi_{k,j} - e_j\psi_{k+1,j} = \frac{\sigma^s}{4} \sum_{n=1}^{N} w_n(\psi_{k+1,n} + \psi_{k,n}) + q_{k+1/2,j} \quad (16b)$$

$k = R,R-1,\cdots,1.$ In these equations

$$d_j = \frac{2|\mu_j| + \Delta\sigma^t}{2\Delta} \quad (16c)$$

and

$$e_j = \frac{2|\mu_j| - \Delta\sigma^t}{2\Delta} \quad (16d)$$

and

$$q_{k+1/2,j} = q(x_{k+1/2}, \mu_j) \ . \quad (16e)$$

Equations (16) could be arranged in the matrix form

$$L\underline{\psi}^* = \underline{q}$$

and the solution $\underline{\psi}^*$ found by inverting L. This procedure is not useful, however, because L is usually dense and is of such large dimension for practical reactor problems that present day computers do not have sufficient storage capacity to invert such systems. Instead of the above procedure, an iterative technique is used for reactor problems which solves (16) in the following manner.

For $\mu_j > 0$,

$$d_j\psi_{k+1,j}^{(i+1)} - e_j\psi_{k,j}^{(i+1)} = \frac{\sigma^s}{4}\sum_{n=1}^{N} w_n(\psi_{k+1,n}^{(i)} + \psi_{k,n}^{(i)}) + q_{k+1/2,j} \qquad (17a)$$

and for $\mu_j < 0$,

$$d_j\psi_{k,j}^{(i+1)} - e_j\psi_{k+1,j}^{(i+1)} = \frac{\sigma^s}{4}\sum_{n=1}^{N} w_n(\psi_{k+1,n}^{(i)} + \psi_{k,n}^{(i)}) + q_{k+1/2,j} \qquad (17b)$$

where i is an iteration index. These equations are the basic equations describing the S_N algorithm.

The iterative solution of this system, $\underline{\psi}^{(i+1)}$, approximates the exact matrix solution $\underline{\psi}^*$. The error in this approximation is called iteration error. A practical strategy is to iterate until this error is arbitrarily small.

Calculation Sequence

The convention of performing the calculation spatially in the direction of neutron flow, as pointed out by Bell,[55] has been observed in equations (16) and (17).

Consequently, iterative and round-off errors are attenuated rather than amplified as the calculation proceeds. This is observed from the S_N algorithm in its machine form. That is, for $\mu_j > 0$

$$\psi^{(i+1)}_{k+1,j} = \frac{e_j}{d_j}\psi^{(i+1)}_{k,j} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\psi^{(i)}_{k+1/2,n} + q_{k+1/2,j}\right\} \quad (18a)$$

and for $\mu_j < 0$

$$\psi^{(i+1)}_{k,j} = \frac{e_j}{d_j}\psi^{(i+1)}_{k+1,j} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\psi^{(i)}_{k+1/2,n} + q_{k+1/2,j}\right\}. \quad (18b)$$

The quantities $\frac{e_j}{d_j}$ are less than unity for all j so any numerical errors in a particular angular flux value will be attenuated in calculating the neighboring angular flux value. Amplification of these errors would have occurred had the convention not been observed.

Iterative Matrix Formulation of the S_N Algorithm

Equations (18) with boundary conditions

$$\psi_{1,j} = 0 \quad \text{for } \mu_j > 0 \quad (18c)$$

$$\psi_{R+1,j} = 0 \quad \text{for } \mu_j < 0 \quad (18d)$$

and an initial guess $\psi^{(0)}_{k,j} = 0$ for all k,j is the S_N machine algorithm for the problem described by equations (10) and (11).

Equations (17) are precisely the same S_N algorithm expressed in slightly different form. If these equations

for an $R+1$ by N mesh (including boundary points) are written so that the solution set is arranged in the following vector order

$$\underline{\psi} = \begin{bmatrix} \psi_{2,1} \\ \psi_{3,1} \\ \vdots \\ \psi_{R+1,1} \\ \psi_{2,2} \\ \psi_{3,2} \\ \vdots \\ \psi_{R+1,2} \\ \vdots \\ \psi_{R+1,N/2} \\ \psi_{R,N/2+1} \\ \psi_{R-1,N/2+1} \\ \vdots \\ \psi_{1,N/2+1} \\ \psi_{R,N/2+2} \\ \psi_{R-1,N/2+2} \\ \vdots \\ \psi_{1,N/2+2} \\ \vdots \\ \psi_{1,N} \end{bmatrix} , \tag{19}$$

which has dimension $R \cdot N$, then equations (17) describe the matrix system

$$(D-E)\underline{\psi}^{(i+1)} = S\,\underline{\psi}^{(i)} + \underline{q}\ . \tag{20}$$

The matrix (D-E) is block diagonal of the form

$$\begin{bmatrix} \boxed{B_1} & & \bigcirc \\ & \boxed{B_2} & \\ & & \ddots & \\ \bigcirc & & & \boxed{B_N} \end{bmatrix} \tag{21}$$

where each block is an R by R submatrix. R is the number of spatial increments in the mesh. There are N such blocks. Each block B_j has the lower bi-diagonal form

$$\begin{bmatrix} d_j & & & & \bigcirc \\ -e_j & d_j & & & \\ & -e_j & d_j & & \\ & & \ddots & \ddots & \\ \bigcirc & & & -e_j & d_j \end{bmatrix} \tag{22}$$

where j is the index belonging to μ_j in the direction set from which e_j and d_j are calculated by equations (16c) and (16d).

The S matrix also has a regular block structure of the form

$$
S = \begin{array}{|c|c|c|c|c|c|c|c|}
\hline
S_{BD_1} & S_{BD_2} & \cdots & S_{BD_{N/2}} & S_{CD_{\frac{N}{2}+1}} & \cdots & S_{CD_{N-1}} & S_{CD_N} \\
\hline
S_{BD_1} & S_{BD_2} & \cdots & S_{BD_{N/2}} & S_{CD_{\frac{N}{2}+1}} & \cdots & S_{CD_{N-1}} & S_{CD_N} \\
\hline
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\hline
S_{BD_1} & S_{BD_2} & \cdots & S_{BD_{N/2}} & S_{CD_{\frac{N}{2}+1}} & \cdots & S_{CD_{N-1}} & S_{CD_N} \\
\hline
S_{CD_1} & S_{CD_2} & \cdots & S_{CD_{N/2}} & S_{BD_{\frac{N}{2}+1}} & \cdots & S_{BD_{N-1}} & S_{BD_N} \\
\hline
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\hline
S_{CD_1} & S_{CD_2} & \cdots & S_{CD_{N/2}} & S_{BD_{\frac{N}{2}+1}} & \cdots & S_{BD_{N-1}} & S_{BD_N} \\
\hline
S_{CD_1} & S_{CD_2} & \cdots & S_{CD_{N/2}} & S_{BD_{\frac{N}{2}+1}} & \cdots & S_{BD_{N-1}} & S_{BD_N} \\
\hline
\end{array} \qquad (23)
$$

where the notation S_{CD} and S_{BD} distinguishes the form of the block submatrix and the subscript denotes the index belonging to all elements in a particular block. The S matrix has certain block regularities listed below:

1) The same index appears in each column of blocks.

2) Each block is an R by R submatrix of S.

3) There are N blocks in each row and column of blocks.

4) The upper right and lower left quadrants of S are composed of blocks, each with form S_{CD}.

5) The upper left and lower right quadrants of S are composed of blocks, each with the form S_{BD}.

The block S_{BD_i} form is

$$\begin{bmatrix} s_i & & & & & \\ s_i & s_i & & & \bigcirc & \\ & s_i & s_i & & & \\ & & \ddots & \ddots & & \\ & \bigcirc & & \ddots & \ddots & \\ & & & & s_i & s_i \end{bmatrix} \tag{24}$$

where each matrix element has the same value

$$s_i = \frac{\sigma^s}{4} w_i \tag{25}$$

and w_i is the i^{th} weight in the quadrature set. Observe that S_{BD_i} is lower bi-diagonal.

The block S_{CD_i} form is

$$\begin{bmatrix} & & & & s_i & s_i \\ & \bigcirc & & s_i & s_i & \\ & & \cdot & \cdot & s_i & \\ & & \cdot & \cdot & & \\ & \cdot & \cdot & & \bigcirc & \\ s_i & \cdot & & & & \\ s_i & & & & & \end{bmatrix} \tag{26}$$

which will be called upper cross bidiagonal. The matrix elements are given by (25).

Inverting the Matrix (D-E)

Due to the simple block form of (D-E) and the fact that each block is lower bidiagonal, see (21) and 22), its inverse is of a simple form. In fact $(D-E)^{-1}$ is of the block form

$$\begin{bmatrix} B_1^{-1} & & & O \\ & B_2^{-1} & & \\ & & \ddots & \\ O & & & B_N^{-1} \end{bmatrix} \tag{27}$$

where each block, B_i, of (21) is inverted individually and placed in the same position in $(D-E)^{-1}$ that it held in (D-E).

Each block B_i^{-1} can easily be calculated from (22) using the method described in Wylie.[57] B_i^{-1} has the form

$$\begin{array}{l} \\ \\ \\ \\ k^{th}\ row \\ \\ \\ \end{array}\left[\begin{array}{ccccccc} \frac{1}{d_i} & & & & & \bigcirc & \\ \frac{e_i}{d_i^2} & \frac{1}{d_i} & & & & & \\ \frac{e_i^2}{d_i^3} & \frac{e_i}{d_i^2} & \frac{1}{d_i} & \ddots & & & \\ \vdots & & & & & & \\ \frac{e_i^{k-1}}{d_i^k} & \frac{e_i^{k-2}}{d_i^{k-1}} & \frac{e_i^{k-3}}{d_i^{k-2}} & \ddots & & & \\ \vdots & & & & \ddots & & \\ \frac{e_i^{R-1}}{d_i^R} & \frac{e_i^{R-2}}{d_i^{R-1}} & \frac{e_i^{R-3}}{d_i^{R-2}} & \cdots & \frac{e_i^2}{d_i^3} & \frac{e_i}{d_i^2} & \frac{1}{d_i} \end{array}\right] = B_i^{-1} \qquad (28)$$

which is lower triangular with elements given by (16c) and (16d).

Since $(D-E)^{-1}$ and S are now known, equation (20) can be operated on by $(D-E)^{-1}S$ to obtain

$$\underline{\psi}^{(i+1)} = (D-E)^{-1}S\ \underline{\psi}^{(i)} + (D-E)^{-1}\ \underline{q}\ . \qquad (29)$$

The matrix $(D-E)^{-1}S$ is called the iteration matrix and can be determined from (23), (24), (26), (27), and (28). $(D-E)^{-1}S$ can be formed by multiplying (27) onto (23) block by block: That is, due to the block diagonal form of $(D-E)^{-1}$, $(D-E)^{-1}S$ is formed by matrix multiplying B_i^{-1} onto each block in the i^{th} row of blocks in S. That is,

$$(D-E)^{-1}S = \begin{array}{|c|c|c|c|c|c|c|c|}
\hline
B_1^{-1}S_{BD_1} & B_1^{-1}S_{BD_2} & \cdots & B_1^{-1}S_{BD_{\frac{N}{2}}} & B_1^{-1}S_{CD_{\frac{N}{2}+1}} & \cdots & B_1^{-1}S_{CD_{N-1}} & B_1^{-1}S_{CD_N} \\
\hline
B_2^{-1}S_{BD_1} & B_2^{-1}S_{BD_2} & \cdots & B_2^{-1}S_{BD_{\frac{N}{2}}} & B_2^{-1}S_{CD_{\frac{N}{2}+1}} & \cdots & B_2^{-1}S_{CD_{N-1}} & B_2^{-1}S_{CD_N} \\
\hline
\vdots & & & \vdots & \vdots & & \vdots & \vdots \\
\hline
B_{\frac{N}{2}}^{-1}S_{BD_1} & B_{\frac{N}{2}}^{-1}S_{BD_2} & \cdots & B_{\frac{N}{2}}^{-1}S_{BD_{\frac{N}{2}}} & B_{\frac{N}{2}}^{-1}S_{CD_{\frac{N}{2}+1}} & \cdots & B_{\frac{N}{2}}^{-1}S_{CD_{N-1}} & B_{\frac{N}{2}}^{-1}S_{CD_N} \\
\hline
B_{\frac{N}{2}+1}^{-1}S_{CD_1} & B_{\frac{N}{2}+1}^{-1}S_{CD_2} & \cdots & B_{\frac{N}{2}+1}^{-1}S_{CD_{\frac{N}{2}}} & B_{\frac{N}{2}+1}^{-1}S_{BD_{\frac{N}{2}+1}} & \cdots & B_{\frac{N}{2}+1}^{-1}S_{BD_{N-1}} & B_{\frac{N}{2}+1}^{-1}S_{BD_N} \\
\hline
\vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
\hline
B_N^{-1}S_{CD_1} & B_N^{-1}S_{CD_2} & \cdots & B_N^{-1}S_{CD_{\frac{N}{2}}} & B_N^{-1}S_{BD_{\frac{N}{2}+1}} & \cdots & B_N^{-1}S_{BD_{N-1}} & B_N^{-1}S_{BD_N} \\
\hline
\end{array} \qquad (30)$$

where each block is of dimension R by R. There are N by N such blocks in $(D-E)^{-1}S$. The following regularities in the block structure of $(D-E)^{-1}S$ are observed:

1) Each block in the i^{th} row of blocks has B_i^{-1} as an operator where i is the index for μ_i.

2) Each block in the j^{th} column of blocks has either S_{BD_j} or S_{CD_j} as an operator where j is the index on ω_j.

3) The upper-right and lower-left quadrants of $(D-E)^{-1}S$ are composed of blocks, each of which includes an operator of the form S_{CD_j}.

4) The upper left and lower right quadrants of $(D-E)^{-1}S$ are composed of blocks, each of which includes an operator of the form S_{BD_j}.

The matrix form of each block of $(D-E)^{-1}S$ is one of two kinds, $B_iS_{BD_j}$ or $B_iS_{CD_j}$.

The form of $B_iS_{BD_j}$ is

$$
\frac{\omega_j \sigma^s}{4} \cdot
\left|\begin{array}{c|c|c|c|c|c|c}
\frac{1}{d_i} & & & & & & \\
\frac{e_i}{d_i^2} + \frac{1}{d_i} & \frac{1}{d_i} & & & & & \\
\frac{e_i^2}{d_i^3} + \frac{e_i}{d_i^2} & \frac{e_i}{d_i^2} + \frac{1}{d_i} & \frac{1}{d_i} & & & \bigcirc & \\
\vdots & \vdots & \vdots & \ddots & & & \\
\frac{e_i^{R-3}}{d_i^{R-2}} + \frac{e_i^{R-4}}{d_i^{R-3}} & \frac{e_i^{R-4}}{d_i^{R-3}} + \frac{e_i^{R-5}}{d_i^{R-4}} & \frac{e_i^{R-5}}{d_i^{R-4}} + \frac{e_i^{R-6}}{d_i^{R-5}} & \cdots & \frac{1}{d_i} & & \\
\frac{e_i^{R-2}}{d_i^{R-1}} + \frac{e_i^{R-3}}{d_i^{R-2}} & \frac{e_i^{R-3}}{d_i^{R-2}} + \frac{e_i^{R-4}}{d_i^{R-3}} & \frac{e_i^{R-4}}{d_i^{R-3}} + \frac{e_i^{R-5}}{d_i^{R-4}} & \cdots & \frac{e_i}{d_i^2} + \frac{1}{d_i} & \frac{1}{d_i} & \\
\frac{e_i^{R-1}}{d_i^{R}} + \frac{e_i^{R-2}}{d_i^{R-1}} & \frac{e_i^{R-2}}{d_i^{R-1}} + \frac{e_i^{R-3}}{d_i^{R-2}} & \frac{e_i^{R-3}}{d_i^{R-2}} + \frac{e_i^{R-4}}{d_i^{R-3}} & \cdots & \frac{e_i^2}{d_i^3} + \frac{e_i}{d_i^2} & \frac{e_i}{d_i^2} + \frac{1}{d_i} & \frac{1}{d_i}
\end{array}\right|
= B_i^{-1} S_{BD_j} \qquad (31)
$$

where e_i and d_i are values of equations (16c) and (16d) evaluated at $|\mu_i|$ and ω_j is the j^{th} weight in the quadrature set.

The form of $B_i^{-1} S_{CD_j}$ is

$$
\frac{\omega_j \sigma^s}{4} \cdot
\left[\begin{array}{c|c|c|c|c|c|c}
 & & & & & \frac{1}{d_i} & \frac{1}{d_i} \\
 & & & & \frac{1}{d_i} & \frac{1}{d_i}+\frac{e_i}{d_i^2} & \frac{e_i}{d_i^2} \\
 & & \bigcirc & \cdot^{\cdot^{\cdot}} & \frac{1}{d_i}+\frac{e_i}{d_i^2} & \frac{e_i}{d_i^2}+\frac{e_i^2}{d_i^3} & \frac{e_i^2}{d_i^3} \\
 & & & \cdot^{\cdot^{\cdot}} & \vdots & \vdots & \vdots \\
 & & \frac{1}{d_i} & & & & \\
 & \frac{1}{d_i} & \frac{1}{d_i}+\frac{e_i}{d_i^2} & \cdots & \frac{e_i^{R-5}}{d_i^{R-4}}+\frac{e_i^{R-4}}{d_i^{R-3}} & \frac{e_i^{R-4}}{d_i^{R-3}}+\frac{e_i^{R-3}}{d_i^{R-2}} & \frac{e_i^{R-3}}{d_i^{R-2}} \\
\frac{1}{d_i} & \frac{1}{d_i}+\frac{e_i}{d_i^2} & \frac{e_i}{d_i^2}+\frac{e_i^2}{d_i^3} & \cdots & \frac{e_i^{R-4}}{d_i^{R-3}}+\frac{e_i^{R-3}}{d_i^{R-2}} & \frac{e_i^{R-3}}{d_i^{R-2}}+\frac{e_i^{R-2}}{d_i^{R-1}} & \frac{e_i^{R-2}}{d_i^{R-1}} \\
\frac{1}{d_i}+\frac{e_i}{d_i^2} & \frac{e_i}{d_i^2}+\frac{e_i^2}{d_i^3} & \frac{e_i^2}{d_i^3}+\frac{e_i^3}{d_i^4} & \cdots & \frac{e_i^{R-3}}{d_i^{R-2}}+\frac{e_i^{R-2}}{d_i^{R-1}} & \frac{e_i^{R-2}}{d_i^{R-1}}+\frac{e_i^{R-1}}{d_i^{R}} & \frac{e_i^{R-1}}{d_i^{R}}
\end{array}\right]
= B_i S_{CD_j} \tag{32}
$$

The fact that there are, at most, two terms in each element of $(D-E)^{-1}S$ is attributed to the form of S_{BD_j} and S_{CD_j} each of which have, at most, two non-zero elements in each column. These properties give an ordered regularity to each block of $(D-E)^{-1}S$.

Equivalence of S_N Algorithm to Matrix Formulation

As far as the author can determine, the S_N algorithm has not previously been described explicitly in the matrix form of the preceding section. The discovery that the S_N algorithm, described by equations (18) is equivalent to solving matrix equation (29) at each iterate step should be no surprise since equations (18) were derived directly from equations (17). At each iterative step the S_N algorithm provides a recipe for inverting $(D-E)$ and multiplying the result into $S\underline{\psi}^{(i)}$ and $\underline{q}$.

This discovery, however, allows the S_N algorithm to be classified as a method of successive approximation which provides a mathematical description of its convergence properties and error analysis. That is, equation (29) is precisely the same as (6). Consequently, properties previously discussed for the method of successive approximations apply to the S_N method if the conditions of Theorem 1 are met. Before this is attempted, it is necessary to show that the norm of the iteration matrix can be calculated for the S_N problem previously described.

Calculation of $||(D-E)^{-1}S||$

The norm imposed upon the system is the infinity norm, defined for a vector by

$$||\underline{\psi}||_\infty = \max_k |\psi_k| \ . \tag{33}$$

Obviously, this norm is conveniently chosen for pointwise error analysis. Isaacson[53] shows that for a matrix, A,

$$||A||_\infty = \max_i \sum_j |a_{ij}| \ , \tag{34}$$

that is, the maximum row sum of absolute values of the matrix elements. From this point, the above definitions will hold for the norms simply denoted by $||\ \ ||$.

The row sum of $(D-E)^{-1}S$, described by (30), (31), and (32) will be a row sum in a row of blocks which comprise $(D-E)^{-1}S$. Since each row of blocks has the same form, it is sufficient to sum a row of blocks and pick the

row from the block sum which gives the maximum row sum of absolute values for $(D-E)^{-1}S$.

Due to the ordering of the direction set,

$$w_{\frac{N}{2}+j} = w_j \tag{35}$$

for $j = 1,2,\cdots,N/2$. A property of the direction set is that

$$\sum_{j=1}^{N/2} w_j = 1 . \tag{36}$$

Consequently a block row sum of $(D-E)^{-1}S$ can be collapsed by taking

$$B_i^{-1}S_{BD_1} + B_i^{-1}S_{BD_2} + \ldots + B_i^{-1}S_{CD_N} =$$

$$= (B_i^{-1}S_{BD_1} + B_i^{-1}S_{CD_{\frac{N}{2}+1}}) + (B_i^{-1}S_{BD_2} + B_i^{-1}S_{CD_{\frac{N}{2}+2}}) + \ldots + (B_i^{-1}S_{BD_{\frac{N}{2}}} + B_i^{-1}S_{CD_N})$$

$$= \frac{\sigma^s}{4}\left\{w_1(B_i^{-1}S_{BD} + B_i^{-1}S_{CD}) + w_2(B_i^{-1}S_{BD} + B_i^{-1}S_{CD}) + \ldots + w_{\frac{N}{2}}(B_i^{-1}S_{BD} + B_i^{-1}S_{CD})\right\}$$

$$= \frac{\sigma^s}{4}(w_1 + w_2 + \ldots + w_{\frac{N}{2}})(B_i^{-1}S_{BD} + B_i^{-1}S_{CD})$$

$$= \frac{\sigma^s}{4}(B_i^{-1}S_{BD} + B_i^{-1}S_{CD}) \tag{37}$$

where $B_i^{-1}S_{BD} + B_i^{-1}S_{CD}$ = sum of the matrix parts of equations (31) and (32).

It is observed from (37), (31), and (32) that the sum of absolute values of the elements of each successive row include all elements of the preceding row plus a positive

quantity. This is true of all row sums except the last. The last row sum is equal to

$$\frac{\sigma^s}{4}\left[3\left(\left|\frac{1}{d_i}\right| + \left|\frac{e_i^{R-1}}{d_i^R}\right|\right) + 4\left(\left|\frac{e_i}{d_i^2}\right| + \left|\frac{e_i^2}{d_i^3}\right| + \cdots + \left|\frac{e_i^{R-2}}{d_i^{R-1}}\right|\right)\right] . \tag{38}$$

The next to last row sum is equal to

$$\frac{\sigma^s}{4}\left[3\left|\frac{e_i^{R-2}}{d_i^{R-1}}\right| + 4\left(\left|\frac{1}{d_i}\right| + \left|\frac{e_i}{d_i^2}\right| + \left|\frac{e_i^2}{d_i^3}\right| + \cdots + \left|\frac{e_i^{R-3}}{d_i^{R-2}}\right|\right)\right] . \tag{39}$$

The last row sum is the maximum if,

$$\left|\frac{e_i^{R-2}}{d_i^{R-1}}\right| + 3\left|\frac{e_i^{R-1}}{d_i^R}\right| > \left|\frac{1}{d_i}\right| \quad \text{or if} \quad \left|\frac{e_i}{d_i}\right|^{R-2} + 3\left|\frac{e_i}{d_i}\right|^{R-1} > 1 . \tag{40}$$

Even if Δ, σ^t, and $|\mu_i|$ were adjusted so that $e_i < 0$, the quotient $\frac{|e_i|}{d_i}$ is less than unity because $|e_i| < d_i$ for all i. Consequently, for systems with large R, the next to last row sum is the maximum.

Now the block row sum must be picked from all possibilities. The index i for which expressions (38) and (39) are maximum can be determined from equations (16c) and (16d).

The quantity

$$\frac{|e_i|}{d_i} = \frac{\left|1 - \frac{\Delta\sigma^t}{2|\mu_i|}\right|}{1 + \frac{\Delta\sigma^t}{2|\mu_i|}}$$

attains its smallest value for a given Δ and σ^t when $|\mu_i|$ is the minimum in the quadrature set $\{\mu_j\}$. Also

$\frac{1}{d_i} = \frac{2\Delta}{2|\mu_i| + \Delta\sigma^t}$ is a maximum for the same $|\mu_i|$. The latter factor dominates, however, in the norm determination. Therefore the norm is calculated by using the minimum absolute ordinate value in the set $\{\mu_j\}$. Consequently

$$||(D-E)^{-1}S|| \le \left\{ \begin{array}{ll} \frac{\sigma^s}{4}\left[\frac{3}{d_i}\left(1+\left|\frac{e_i}{d_i}\right|^{R-1}\right)+\frac{4}{d_i}\left(\left|\frac{e_i}{d_i}\right|+\left|\frac{e_i}{d_i}\right|^2+\ldots+\left|\frac{e_i}{d_i}\right|^{R-2}\right)\right] & if \left|\frac{e_i}{d_i}\right|^{R-2}+3\left|\frac{e_i}{d_i}\right|^{R-1} > 1 \\ \frac{\sigma^s}{4}\left[\frac{3}{d_i}\left|\frac{e_i}{d_i}\right|^{R-2}+\frac{4}{d_i}\left(1+\left|\frac{e_i}{d_i}\right|+\left|\frac{e_i}{d_i}\right|^2+,\ldots+\left|\frac{e_i}{d_i}\right|^{R-3}\right)\right] & \text{otherwise} \end{array} \right\} \quad (41)$$

where i is the index for the minimum absolute ordinate value in the set $\{\mu_j\}$. The latter value is appropriate for large R. Observe that equality occurs when $e_i \ge 0$.

For a given $\Delta, \sigma^t, \sigma^s$, $|\mu_i|_{min}$, and R the infinity-norm of the iteration matrix can be calculated by (41). It is obvious that the norm is bounded since $\frac{|e_i|}{d_i} < 1$ and $\frac{1}{d_i}$ is bounded for any finite dimensional system.

Observe that, for a fixed Δ, σ^t, and $|\mu_i|_{min}$, the coefficients e_j and d_j are fixed. In this case the norm decreases linearly with σ^s. Consequently the norm of the iteration matrix is small for problems involving primarily absorbing media and large near unity for predominantly scattering media. This results in rapid convergence to an acceptable iterative error for absorbing media and slow convergence to the same level of error for predominantly scattering media. This behavior is predicted by equation (9). It will be verified experimentally later.

Sufficient Conditions for Convergence of S_N

Equation (41) will be used to provide sufficient conditions to guarantee that $||(D-E)^{-1}S|| < 1$ in order that the S_N algorithm can meet the conditions of Theorem 1. For large R, $||(D-E)^{-1}S||$ is evaluated by the second expression in (41), which can be rewritten

$$||(D-E)^{-1}S|| \leq \frac{\sigma^s}{4d_i}\left[4\left(\frac{1-\beta^{R-1}}{1-\beta}\right) - \beta^{R-2}\right] \tag{42}$$

where $\quad 0 \leq \beta = \dfrac{|e_i|}{d_i} < 1 .$ (43)

For large R the terms β^{R-1} and β^{R-2} are insignificant and could be neglected. But since

$$4\left(\frac{1-\beta^{R-1}}{1-\beta}\right) - \beta^{R-2} < \frac{4}{1-\beta} ,$$

imposing the condition

$$\frac{\sigma^s}{d_i}\left(\frac{1}{1-\beta}\right) < 1 \tag{44}$$

will guarantee that $||(D-E)^{-1}S|| < 1$ without seriously loosening the resulting bounds on Δ.

From (16c), (16d), and (43), expression (44) becomes

$$d_i > \sigma^s + |e_i|$$

or

$$2|\mu_i|_{min} + \Delta\sigma^t > \Delta\sigma^s + \left|2|\mu_i|_{min} - \Delta\sigma^t\right| . \tag{45}$$

This inequality provides the sufficiency conditions for convergence of the S_N algorithm.

Suppose the problem is defined so that σ^s, σ^t, and $|\mu_i|_{min}$ have been selected and it is desired to impose a sufficient condition on the choice of Δ so that (45) is satisfied. Then the condition $||(D-E)^{-1}S|| < 1$ is satisfied.

Suppose further that $\Delta\sigma^t > 2|\mu_i|_{min}$ so that $e_i < 0$. Then (45) becomes

$$2|\mu_i|_{min} + \Delta\sigma^t > 2\Delta\sigma^s + \Delta\sigma^t - 2|\mu_i|_{min}$$

or

$$\Delta < \frac{2|\mu_i|_{min}}{\sigma^s} \quad . \tag{46}$$

This imposes a limit on the negativity of the elements e_i.

If $\Delta\sigma^t$ had been chosen equal to $2|\mu_i|_{min}$. Then $\beta=0$ in (42) and $\frac{\sigma^s}{d_i} < 1$ is a sufficient condition for $||(D-E)^{-1}S|| < 1$. But this leads to the inequality

$$\Delta < \frac{2|\mu_i|_{min}}{2\sigma^s - \sigma^t} \quad . \quad \text{But } \Delta = \frac{2|\mu_i|_{min}}{\sigma^t}$$

so the inequality becomes $\frac{1}{\sigma^t} < \frac{1}{2\sigma^s - \sigma^t}$ or $\sigma^s < \sigma^t$. Consequently $||(D-E)^{-1}S|| < 1$ in this case if

$$0 \le \frac{\sigma^s}{\sigma^t} < 1 \quad . \tag{47}$$

Finally, suppose $\Delta\sigma^t < 2|\mu_i|_{min}$ is selected. This choice guarantees that e_j and d_j are positive for all j, hence $(D-E)^{-1}S$ and $(D-E)^{-1}$ are positive operators. Expression (45) is then

$$2|\mu_i|_{min} + \Delta\sigma^t > 2\Delta\sigma^s + 2|\mu_i|_{min} - \Delta\sigma^t$$

or

$$\frac{\sigma^s}{\sigma^t} < 1 . \tag{48}$$

Consequently, $||(D-E)^{-1}S|| < 1$ for all physical problems with this choice of Δ.

Properties of $(D-E)^{-1}$

The matrix $(D-E)$ exists in the construction given by (21) and (22). It is a linear operator due to its construction from the discretized linear equations (17). It is a bounded operator since

$$||(D-E)|| = d_i + |e_i|$$

which is bounded for all finite values of $\sigma^s, \sigma^t, |\mu_i|_{min}$, and Δ.

The matrix $(D-E)^{-1}$ exists if $det(D-E)$ is non-zero. From equations (21) and (22)

$$det(D-E) = (d_1 d_2 \cdots d_N)^R$$

and since $d_j > 0$ for all j, $det(D-E) \neq 0$.

All conditions of Theorem 1 are met by the S_N algorithm if the sufficiency conditions listed in Table I are met by the problem parameters. If these conditions are met, $||(D-E)^{-1}S|| < 1$ and the S_N algorithm converges to $\underline{\psi}^*$, the exact solution of the discretized set of equations.

TABLE I

SUFFICIENCY CONDITIONS FOR CONVERGENCE OF S_N

Possible Choice For Δ	Value of e_i	Bound on Δ to Meet Sufficiency Condition	Values of σ^s/σ^t for Which S_N Algorithm Converges
$\Delta > \frac{2\|\mu_i\|min}{\sigma^t}$	Negative	$\Delta < \frac{2\|\mu_i\|min}{\sigma^s}$	$0 \leq \frac{\sigma^s}{\sigma^t} < 1$
$\Delta = \frac{2\|\mu_i\|min}{\sigma^t}$	0	———	$0 \leq \frac{\sigma^s}{\sigma^t} < 1$
$\Delta < \frac{2\|\mu_i\|min}{\sigma^t}$	Positive	$\Delta < \frac{2\|\mu_i\|min}{\sigma^t}$	$0 \leq \frac{\sigma^s}{\sigma^t} < 1$

The above table can be summarized by stating that the sufficiency condition is met for any Δ provided that $\sigma^s < \sigma^t$ or $\frac{2|\mu_i|min}{\Delta}$, whichever is smaller.

The consequences of the S_N algorithm meeting the sufficiency conditions for convergence are that expressions (1) through (9) can be used to study its convergence and iteration error properties.

Positivity

Positivity of the solution of the S_N equations for positive source and non-negative boundary conditions is desirable because it is a numerical approximation to a non-negative physical quantity. In practice the S_N algorithm has produced negative solutions.[8] One-dimensional discrete ordinates codes such as ANISN[58] and DTF-IV[59] have flux fix-up routines incorporated so that negative solution values are not allowed to occur. This strategy, although of practical merit, is not mathematically rigorous. The effect of these arbitrary fix-ups on the iteration error is not clear. Lathrop[60] investigated various two-dimensional difference schemes and found that the quest for positivity is made at the expense of accuracy and efficiency of convergence. Bell[55] points out that the S_N difference equations may lead to negative solution values since they do not correspond to a positive operator.

The preceding section revealed that sufficiency conditions for convergence of the S_N algorithm can be met which allow the iteration matrix to possess negative matrix elements. But convergence to the exact solution cannot be guaranteed if Δ is chosen so that $\Delta > \frac{2|\mu_i|_{min}}{\sigma^s}$. Consequently, there is a limit on the negativity of some of the elements of $(D-E)^{-1}S$ if the theory is to be invoked to guarantee convergence.

For computational economy reasons it is desirable to choose Δ as large as possible. For the problem

previously described, however, $\Delta < \frac{2|\mu_i|_{min}}{\sigma^s}$ imposes an absolute upper bound on Δ, above which the sufficient condition for convergence is not met. If this bound on Δ is observed, it is reasonable to assume that the approximate solution $\underline{\psi}^{(i)}$ will be non-negative if i is large enough even though initial iterates may have some negative components. This is because $\underline{\psi}^*$ is expected to be a non-negative vector due to its approximation of $\psi_j(x)$ which is non-negative by its physical interpretation and $\underline{\psi}^{(i)}$ approaches $\underline{\psi}^*$ as i becomes large. This is true even though the theory of positive operators, which is used successfully in multi-group diffusion theory,[47] is not applicable in this instance. Of course no guarantee exists that $\underline{\psi}^*$, the discrete approximation to $\psi_j(x)$ is non-negative. Numerical experiments by the author confirm that it is for uniform and first collision sources when the sufficiency conditions are met.

If Δ is chosen so that $\Delta < \frac{2|\mu_i|_{min}}{\sigma^t}$, however, $(D-E)^{-1}S$ and $(D-E)^{-1}$ are non-negative matrix operators which guarantee a non-negative approximation $\underline{\psi}^{(i)}$ for all i.

Necessary Conditions

The previously discussed conditions are not necessary for the S_N iterative process to terminate based upon fractional differences reaching an arbitrarily small value. Indeed termination of the S_N algorithm iterative process has been observed by this investigator for choices of

$\Delta >> \frac{2|\mu_i|min}{\sigma^s}$ for which $||(D-E)^{-1}S|| >> 1$. For some sources, e.g. uniform source, convergence to a positive solution vector occurred. For other sources, e.g. first collision source, the solution vector contained negative components. It is doubtful whether this process can accurately be called convergence since the nature of the resulting solution is not clearly defined.

Even if Δ is chosen so that $\Delta > \frac{2|\mu_i|max}{\sigma^t}$, in which case all e_j are negative, $(D-E)^{-1}S$ contains even and odd powers of e_i/d_i. Consequently, the iteration matrix is not wholly negative. In this instance the S_N algorithm may or may not converge to a non-positive solution depending upon the source distribution. For these cases, where some elements of the solution are negative, doubt exists as to the nature of the solution.

It is not clear that necessary conditions on Δ exist for convergence of the S_N algorithm. It is clear, however, that a solution exists if the sufficiency conditions are met and that these conditions allow a limited amount of negativity for the iteration matrix. For the numerical problems previously cited, these solutions are positive.

Effect of Δ on $||(D-E)^{-1}S||$

Figure 2.2 reveals the effect of Δ on $||(D-E)^{-1}S||$ for the following problem parameters:

slab width = 10.0 cm

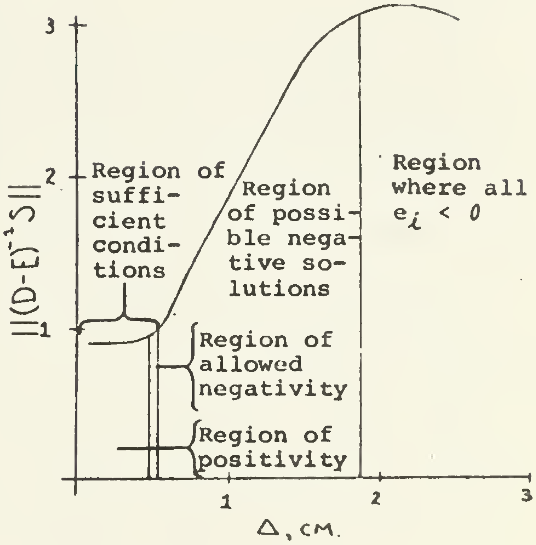
σ^t = 1.0 cm^{-1}

σ^s = 0.9 cm^{-1}

$|\mu_i|_{min}$ = 0.23862

FIGURE 2.2

NORM BEHAVIOR WITH Δ



As R (the number of spatial increments) decreases, Δ becomes larger and $||(D-E)^{-1}S||$ increases. In the region where Δ meets the sufficiency conditions, $||(D-E)^{-1}S|| < 1$ and convergence of S_N to the exact matrix solution is assured. A small part of this region allows negative elements in $(D-E)^{-1}S$.

In the region where $||(D-E)^{-1}S|| > 1$, termination of the iterative process may occur. Solution vectors may be positive or partially negative depending upon the source distribution. This region should be avoided for practical problems unless other evidence is available that the solution has validity.

Effect of Quadrature Set on $||(D-E)^{-1}S||$

Expression (9) reveals that the iterative error is less tightly bounded as $||(D-E)^{-1}S||$ increases towards unity. This means that more iterations are required to

reach a specified accuracy. Equation (41) shows that $||(D-E)^{-1}S||$ increases for fixed $\Delta, \sigma^t, \sigma^s$, as the ordinate $|\mu_i|_{min}$ decreases. This is because $\frac{1}{d_i}$ grows larger and $|e_i|/d_i$ diminishes as Δ increases but $1/d_i$ predominates in (41) so $||(D-E)^{-1}S||$ increases. All quadrature sets have the property that $|\mu_i|_{min}$ approaches zero as the order N of S_N is increased. Consequently, the effect of choosing a higher order S_N is to increase $||(D-E)^{-1}S||$, hence requiring a larger number of iterations to convergence. This loss of efficiency might be observed in going from S_2 to S_4 where the spread in $|\mu_i|_{min}$ values is significant but is of no practical importance for higher order S_N where $|\mu_i|_{min}$ values are quite close.

Convergence Properties of S_N Algorithm

The following discussion applies to the S_N algorithm when $\Delta < \frac{2|\mu_i|_{min}}{\sigma^t}$. Then e_j and d_j are positive for all j and $(D-E)^{-1}S$ and $(D-E)^{-1}$ are non-negative. The sufficient condition for convergence is met for all physically realizable problems. The initial guess, $\underline{\psi}^{(0)} = 0$, is a convenient reference so that comparison of convergence efficiency can be made with another method later.

The S_N algorithm, under these conditions, converges in such a way that each vector component approaches its exact value in a monotone non-negative manner. This can be ascertained from applying equation (7) to the difference defined by

$$\underline{\delta}^{(i+1)} = \underline{\psi}^{(i+1)} - \underline{\psi}^{(i)} . \tag{49}$$

Then

$$\underline{\delta}^{(i+1)} = \sum_{m=0}^{i}[(D-E)^{-1}S]^m(D-E)^{-1}\underline{q} - \sum_{m=0}^{i-1}[(D-E)^{-1}S]^m(D-E)^{-1}\underline{q}$$

$$= [(D-E)^{-1}S]^i(D-E)^{-1}\underline{q} \tag{50}$$

which is positive for $\underline{q} > 0$. Consequently, some positive quantity is added to each component of $\underline{\psi}^{(i)}$ to produce the respective component of $\underline{\psi}^{(i+1)}$.

From the definition of $\underline{e}^{(i)}$ given by (8) and $\underline{\psi}^*$ given by (4),

$$\underline{e}^{(i)} = \underline{\psi}^* - \underline{\psi}^{(i)} = \sum_{m=0}^{\infty}[(D-E)^{-1}S]^m(D-E)^{-1}\underline{q} - \sum_{m=0}^{i-1}[(D-E)^{-1}S]^m(D-E)^{-1}\underline{q}$$

$$= [(D-E)^{-1}S]^i\{I+(D-E)^{-1}S+[(D-E)^{-1}S]^2+\cdots\}(D-E)^{-1}\underline{q} \tag{51}$$

which is positive for $\underline{q} > 0$. But

$$\underline{e}^{(i)} - \underline{e}^{(i+1)} = \underline{\psi}^* - \underline{\psi}^{(i)} - \underline{\psi}^* + \underline{\psi}^{(i+1)} = \underline{\delta}^{(i+1)} . \tag{52}$$

Hence

$$\underline{\delta}^{(i+1)} < \underline{e}^{(i)} \tag{53}$$

due to positivity of $\underline{e}^{(i)}$ and $\underline{e}^{(i+1)}$. See Figure 2.3 for a convergence diagram.

The largest absolute error in each successive iterate is monotone decreasing. From equations (51) and (4),

$$\underline{e}^{(i)} = [(D-E)^{-1}S]^{i}\,\underline{\psi}^* . \qquad (54)$$

Consequently,

$$\underline{e}^{(i+1)} = [(D-E)^{-1}S]^{i+1}\underline{\psi}^* = [(D-E)^{-1}S]\underline{e}^{(i)} . \qquad (55)$$

This means that

$$e_k^{(i+1)} = \{(D-E)^{-1}S\underline{e}^{(i)}\}_k \quad \text{for all } k \text{ which}$$

does not imply that $e_k^{(i+1)}$ is some fixed fraction of $e_k^{(i)}$ for all k even though (52) ensures that $e_k^{(i+1)} < e_k^{(i)}$ for all k. That is, it cannot be shown that $e_k^{(i+1)} < \beta\, e_k^{(i)}$ for all i,k where β is a positive constant less than unity. The most that can be inferred from (55) is

$$||\underline{e}^{(i+1)}|| \le ||(D-E)^{-1}S||\cdot||\underline{e}^{(i)}|| . \qquad (56)$$

This means that the maximum error at each iterative step diminishes if $||(D-E)^{-1}S|| < 1$. This condition is met for convergence.

Similarly it can be shown from equation (50) that

$$||\underline{\delta}^{(i+1)}|| \le ||(D-E)^{-1}S||\cdot||\underline{\delta}^{(i)}|| \qquad (57)$$

using the same arguments and with the same consequences discussed above for the error.

It is not of much practical value to know that the error at each iterative step is less than the maximum error at the previous step. It would be of more practical value to know that the error is bounded by some fraction of the difference at each step because the differences can be calculated. Equation (51) is

$$\underline{e}^{(i)} = [(D-E)^{-1}S]^{i}\{I+(D-E)^{-1}S+[(D-E)^{-1}S]^{2}+\cdots\}(D-E)^{-1}\underline{q}$$

$$= \{I+(D-E)^{-1}S+[(D-E)^{-1}S]^{2}+\cdots\}[(D-E)^{-1}S]^{i}(D-E)^{-1}\underline{q}$$

because $[(D-E)^{-1}S]^{i}$ commutes with each term in the braces. So

$$\underline{e}^{(i)} = \{I+(D-E)^{-1}S+[(D-E)^{-1}S]^{2}+\cdots\}\ \underline{\delta}^{(i+1)}$$

due to (50). Consequently,

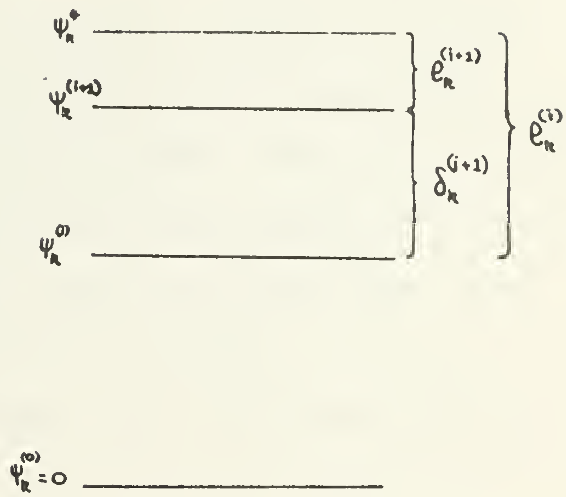
$$e_k^{(i)} \le ||\underline{e}^{(i)}|| \le \frac{||\underline{\delta}^{(i+1)}||}{1-||(D-E)^{-1}S||} \tag{58}$$

for all k. This does not provide enough information to assure that $e_k^{(i)}$ is some fixed fraction of $\delta_k^{(i+1)}$ for all k but it is enough to provide a practical link between the error and difference which can be exploited to derive a useful convergence criterion.

Convergence of the S_N algorithm is pictorially described in Figure 2.3 for each vector component. The iterative solution proceeds progressively towards the exact solution in a monotone positive way governed by properties (53), (56), (57), and (58).

FIGURE 2.3

S_N CONVERGENCE DIAGRAM



Conventional Pointwise Convergence Criterion

The pointwise convergence criterion based on fractional differences,

$$\frac{\psi_k^{(i+1)}-\psi_k^{(i)}}{\psi_k^{(i)}} < \varepsilon \qquad (59)$$

where ε is chosen arbitrarily small, is used universally in practice for shielding problems. It is especially useful in deep penetration problems where $\underline{\psi}^*$ has elements of very small magnitude since it weights these regions more heavily. That is, iteration proceeds until the deep fluxes, which are the last to meet the convergence criterion, are converged. Its utility is based on the practical idea that, as the differences become smaller with successive iterations, the fractional amount added to each solution component is smaller than the desired accuracy after some i. Consequently,

further iteration will not add significantly to the solution.

It must be ascertained, however, whether this convergence criterion, if met, implies that the pointwise fractional error is sufficiently bounded. After all, it is desired to iterate until assured that the pointwise iteration fractional error is arbitrarily small, not just until the pointwise fractional differences are bounded.

Suppose iteration proceeds until condition (59) is met. Since

$$\underline{\delta}^{(i+1)} = \underline{e}^{(i)} - \underline{e}^{(i+1)}$$

this implies that

$$\underline{e}^{(i)} - \underline{e}^{(i+1)} < \varepsilon\, \underline{\psi}^{(i)}$$

but $\underline{e}^{(i+1)} = (D-E)^{-1}S\, \underline{e}^{(i)}$ so the above implies

$$\underline{e}^{(i)} - (D-E)^{-1}S\, \underline{e}^{(i)} < \varepsilon\, \underline{\psi}^{(i)}$$

or

$$\underline{e}^{(i)} < \varepsilon\, \underline{\psi}^{(i)} + (D-E)^{-1}S\, \underline{e}^{(i)} . \tag{60}$$

But by definition

$$[(D-E)^{-1}S\underline{e}^{(i)}]_k \leq ||(D-E)^{-1}S\underline{e}^{(i)}||$$

$$\leq ||(D-E)^{-1}S||\cdot||\underline{e}^{(i)}|| .$$

Consequently (60) implies that

$$e_k^{(i)} < \varepsilon\ \psi_k^{(i)} + ||(D-E)^{-1}S||\cdot||\underline{e}^{(i)}||$$

for all k. But from the definition

$\underline{e}^{(i)} = \underline{\psi}^{*}-\underline{\psi}^{(i)}$, the above inequality becomes

$$e_k^{(i)} < \varepsilon\ [\psi_k^{*}-e_k^{(i)}] + ||(D-E)^{-1}S||\cdot||\underline{e}^{(i)}||$$

or

$$(1+\varepsilon)e_k^{(i)} < \varepsilon\psi_k^{*} + ||(D-E)^{-1}S||\cdot||\underline{e}^{(i)}||$$

for all k. But if this inequality holds for all k, it must hold for j, the value of k which corresponds to the maximum absolute value of $e_k^{(i)}$. Hence

$$||\underline{e}^{(i)}||\cdot\left[1+\varepsilon-||(D-E)^{-1}S||\right] < \varepsilon\ \psi_j^{*}$$

or

$$\frac{||\underline{e}^{(i)}||}{\psi_j^{*}} < \frac{\varepsilon}{1+\varepsilon-||(D-E)^{-1}S||} \tag{61}$$

where j is the index for which $\underline{e}^{(i)}$ attains its maximum. Observe that (61) does not imply that the fractional error is bounded for all components of $\underline{\psi}^{*}$. However, if it is arbitrarily assumed that

$$\psi_k^{*} \geq \psi_j^{*} \tag{62}$$

for all k, expression (61) implies that

$$\frac{||e^{(i)}||}{\psi_k^*} < \frac{\varepsilon}{1+\varepsilon-||(D-E)^{-1}S||}$$

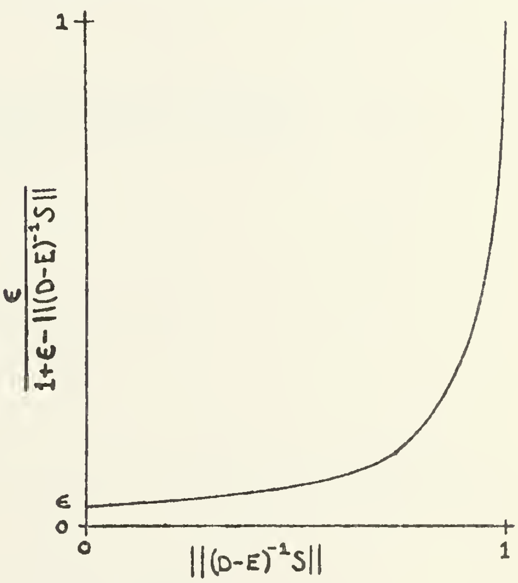
or that

$$\frac{e_k^{(i)}}{\psi_k^*} < \frac{\varepsilon}{1+\varepsilon-||(D-E)^{-1}S||} \tag{63}$$

for all k. But even under the dubious assumption (62), this does not imply a satisfactory bound on the fractional error. Figure 2.4 displays a plot of the bound versus $||(D-E)^{-1}S||$. Observe that, as $||(D-E)^{-1}S||$ approaches unity, the bound grows to 1. That is, up to 100 percent fractional error is implied by invoking (59) as a convergence criterion. This criterion is useful and practical for $||(D-E)^{-1}S|| <$ 0.9 but for larger values it is a poor choice because the resulting bound on the fractional error is too large.

FIGURE 2.4

BOUND ON FRACTIONAL ERROR



An Improved Pointwise Convergence Criterion

Implicit in the concept of a practical convergence criterion is that iteration is terminated only after the fractional error is arbitrarily small. Since the exact solution is not known, it is necessary to iterate on fractional differences, which are known. $||(D-E)^{-1}S||$ can be calculated for the stated S_N problem so it is possible

to establish a convergence criterion based on fractional differences which implies that the fractional errors are arbitrarily small.

Suppose iteration proceeds until

$$\frac{||\underline{\delta}^{(i+1)}||}{\psi_k^{(i)}} < \frac{\varepsilon}{1-\varepsilon}\left[1-||(D-E)^{-1}S||\right] \qquad (64)$$

for all k where ε is some arbitrarily small positive number. This implies that

$$\frac{||\underline{\delta}^{(i+1)}||}{1-||(D-E)^{-1}S||} < \frac{\varepsilon\ \psi_k^{(i)}}{1-\varepsilon} \text{ for all } k.$$

But repeating (58)

$$e_k^{(i)} \leq \frac{||\underline{\delta}^{(i+1)}||}{1-||(D-E)^{-1}S||} \text{ so the above}$$

inequality implies that

$$e_k^{(i)} < \frac{\varepsilon}{1-\varepsilon}\ \psi_k^{(i)} \text{ for all } k.$$

But by definition $\underline{\psi}^{(i)} = \underline{\psi}^* - \underline{e}^{(i)}$, so the above inequality implies that

$$\underline{e}^{(i)} < \frac{\varepsilon}{1-\varepsilon}\ [\underline{\psi}^* - \underline{e}^{(i)}]$$

or

$$\underline{e}^{(i)} < \varepsilon\ \underline{\psi}^*$$

or finally

$$\frac{\psi_k^* - \psi_k^{(i)}}{\psi_k^*} < \varepsilon \qquad (65)$$

for all k. Consequently, upon invoking convergence criterion (64), the convergence properties of the S_N algorithm assure that the fractional iteration error is arbitrarily small. This is precisely the goal for a machine convergence criterion.

For the S_N problem discussed here, equation (41) allows the exact computation of $||(D-E)^{-1}S||$ and ε is arbitrarily chosen. Of course for problems in which $||(D-E)^{-1}S||$ is not obtainable, it is necessary to use (59) as the convergence criterion but recognition of its inability to guarantee a satisfactory bound on the fractional error for problems involving strongly scattering media is imperative.

FIGURE 2.5

BEHAVIOR OF IMPROVED CONVERGENCE CRITERION

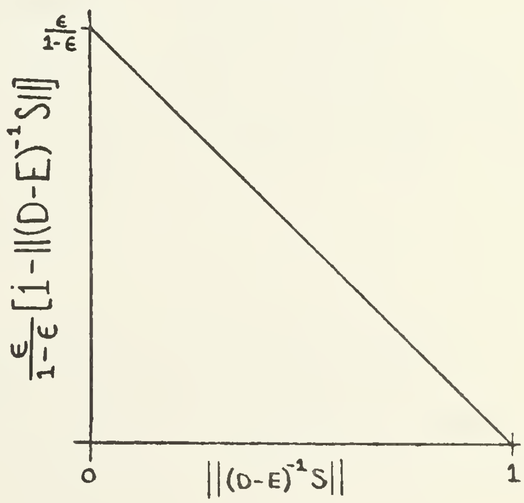


Figure 2.5 reveals that the convergence constant, $\frac{\varepsilon}{1-\varepsilon}\left[1-||(D-E)^{-1}S||\right]$ becomes vanishingly small as $||(D-E)^{-1}S||$ approaches unity. For predominantly scattering media $||(D-E)^{-1}S||$ is quite close to unity. It is necessary to use the improved convergence criterion (64) for these problems. For weakly scattering media, i.e. $\sigma^s/\sigma^t <$ 0.9, the conventional convergence criterion is acceptable. Table II shows that the effect of using the new convergence criterion is small for $||(D-E)^{-1}S|| < 0.9$ but overpowering for $||(D-E)^{-1}S|| > 0.9$.

TABLE II

IMPROVED CONVERGENCE CRITERION
FOR $\varepsilon = 10^{-4}$

$\|\|(D-E)^{-1}S\|\|$	$\frac{\varepsilon}{1-\varepsilon}\left[1-\|\|(D-E)^{-1}S\|\|\right]$
0.5	0.5×10^{-4}
0.9	10^{-5}
0.99	10^{-6}
0.999	10^{-7}

The improved convergence criterion imposes a much more severe condition for convergence than does the conventional convergence criterion, especially for predominantly scattering media. For these problems, convergence is slow even using the conventional criterion. Imposing the improved convergence criterion makes a bad situation worse. It is necessary, however, if assurance is required that the fractional error is arbitrarily small.

Computer Experiment

A sample S_N problem, described by equations (18), was run on a 360/67 IBM computer with the following parameters:

Slab width = 10.0 cm

R = 30 spatial intervals

$\sigma^t = 1.0\ \mathrm{cm}^{-1}$

σ^s = varied

$\varepsilon = 10^{-4}$

N = 6 ordinates using Gauss quadrature set[61]

Uniform source, $q_j(x) = 1.0$ for all j.

It is worthy of note that any of the more recently derived mechanical quadrature sets,[23,62] could have been used but the Gauss set is satisfactory to demonstrate convergence properties of the S_N algorithm. This set will be used throughout.

A copy of the computer program and samples of the computer output are listed in Appendix A. Table III lists the results. The number of iterations to convergence are tabulated for various σ^s/σ^t ratios using each of three different convergence criterion.

Observe that, regardless of the convergence criterion used, problems involving predominantly absorbing media required fewer iterations to converge than did those for scattering media. The number of iterations to convergence is a monotone increasing function of σ^s/σ^t.

Comparing columns two and three shows the effect of converging on the maximum difference instead of the pointwise difference. As expected, the maximum difference criterion is more severe since it takes more iterations to meet it than the pointwise difference criterion for each value of σ^s/σ^t. This is expected because

$$\frac{||\delta^{(i+1)}||}{\psi_k^{(i)}} \geq \frac{\delta_k^{(i+1)}}{\psi_k^{(i)}} \quad \text{for all } k \text{ by definition.}$$

TABLE III

NUMBER OF ITERATIONS TO CONVERGENCE
$\varepsilon = 10^{-4}$

$\frac{\sigma^s}{\sigma^t}$, cm^{-1}	Convergence Criterion			% Increase of Column 4 Over Column 2
	$\frac{\delta_k^{(i+1)}}{\psi_k^{(i)}} < \varepsilon$	$\left\| \frac{\underline{\delta}^{(i+1)}}{\psi_k^{(i)}} \right\| < \varepsilon$	$\left\| \frac{\underline{\delta}^{(i+1)}}{\psi_k^{(i)}} \right\| < \frac{\varepsilon}{1-\varepsilon}\left[1-\|(D-E)^{-1}S\|\right]$	
0.5	14	16	17	21.4
0.9	57	73	91	59.6
0.99	171	242	374	118
0.999	219	317	586	167
0.9999	225	327	699	211

For this particular problem, $\|(D-E)^{-1}S\| = \frac{\sigma^s}{\sigma^t}$.

Comparison of columns two and four shows the effect of the improved convergence criterion which guarantees that the fractional iteration error is less than ε. Its effect includes the effect discussed in the preceding paragraph and has the additional effect of $||(D-E)^{-1}S||$. Both effects increase the number of iterations to convergence. The effect of $||(D-E)^{-1}S||$ becomes more dominant as σ^s/σ^t increases above 0.9. This is due to the direct linear relation σ^s has on $||(D-E)^{-1}S||$.

A comparison of solutions, one of which met convergence criterion (59), the other meeting the improved convergence criterion (64), was made for the values of σ^s/σ^t listed in Table IV.

TABLE IV

COMPARISON OF SOLUTION VALUES
$\varepsilon = 10^{-4}$

σ^s/σ^t	Agreement of Conventional Convergence Criterion Solution with Improved Convergence Criterion Solution
0.5	Agreement in 4th significant figure
0.9	All agree to 2nd significant figure, many to 3rd
0.99	All agree to 2nd significant figure, some to 3rd
0.999	All agree to 2nd significant figure, some to 3rd
0.9999	All agree to 2nd significant figure, few to 3rd

For this problem $||(D-E)^{-1}S|| = \sigma^s/\sigma^t$.

The above table reveals that agreement exists only to the second significant figure for $\sigma^s/\sigma^t > 0.9$. From previous discussion it was established that a more accurate solution results when a larger number of iterations are made. Consequently the improved convergence criterion solution must be closer to the exact solution than the conventional convergence criterion solution because more iterations were performed for the former. The above table shows that the conventional convergence criterion solution agreed to only the second significant figure with the more accurate improved convergence criterion solution. This illustrates that the expected accuracy of $\varepsilon = 10^{-4}$ was not achieved for large values of σ^s/σ^t using the conventional convergence criterion.

Tables III and IV provide vivid illustrations of the necessity for using the improved convergence criterion for problems with $||(D-E)^{-1}S|| > 0.9$ if an arbitrarily small fractional iterative error is required.

The overall effect of using the improved convergence criterion is insignificant for problems involving strongly absorbing media for which $||(D-E)^{-1}S||$ is small. The effect is marked, however, for predominantly scattering systems and results in additional computational work required to assure arbitrarily small fractional iterative errors. Consequently any scheme which accelerates convergence of the S_N algorithm by decreasing $||(D-E)^{-1}S||$ would be eminently useful when invoking the new convergence criterion, especially for strong scattering media problems.

CHAPTER III

ACCELERATION OF S_N CONVERGENCE BY SPATIAL TRANSFORMS

General Discussion

The previous chapter discussed how the S_N algorithm converges more rapidly for absorbing than for scattering media. As σ^s decreases, holding σ^t constant, the number of iterations to convergence decreases regardless of the choice of convergence criterion. Based on this physical argument, it is expected that a mathematical device which adds absorption to the problem will accelerate convergence. From a mathematical point of view, an operation on the S_N equations which decreases the norm of the iteration matrix will improve the efficiency of the algorithm. A new method is presented here which incorporates both ideas.

General Spatial Transform

A transform of the type

$$\psi(x) = \phi(x) g^{\alpha}(x) \tag{1}$$

has been used to provide an effective importance sampling device in the solution of the transport equation by Monte Carlo methods.[63] It biases the sampling distribution towards more important directions, thereby reducing the variance in Monte Carlo methods. This leads to improved statistical accuracy for a given calculational effort.

Transforms of this type have been used by the author to accelerate convergence of the S_N algorithm for certain problems. Transforms which have proven successful are:

$$\psi_j(x) = \phi_j(x) e^{\alpha x} , \tag{2}$$

$$\psi_j(x) = \phi_j(x) \tanh^{\alpha} \left(\frac{x+b}{W}\right) , \tag{3}$$

and

$$\psi_j(x) = \phi_j(x) \left[\frac{x+b}{\ln(x+b)}\right]^{\alpha} \tag{4}$$

where α is an arbitrarily chosen acceleration parameter. W and b are parameters chosen to restrict the range of the arguments. These transforms are applied to the angular discretized transport equations. The most effective transform of those listed is (2).

The Half-Range Problem

The spatial transform accelerates convergence of the S_N algorithm for the deep penetration infinite slab problem under the following assumptions:

i) steady state

ii) homogeneous medium

iii) angular flux has no azimuthal dependence

iv) high energy particles are scattered only in forward directions

v) forward scattering is equally distributed in direction.

A consequence of these assumptions is that the transport equation has the form

$$\mu\frac{\partial\psi(x,\mu)}{\partial x} + \sigma^t\psi(x,\mu) = \int_0^1 \sigma^s f(\mu'\rightarrow\mu)\psi(x,\mu')d\mu' + q(x,\mu) \qquad (5)$$

valid for

$$0 < \mu \leq 1 .$$

The anisotropic forward scattering assumption is valid for high energy neutrons and gammas. The assumption that the forward scattered particles are equally distributed in angle is a simplification of the usual Legendre Polynomial approximation and is made strictly for convenience. The more accurate description of the forward scattering anisotropy could be treated by the method.

The scattering distribution function, under assumption v) is normalized by recognizing that $f(\mu'\rightarrow\mu) = const$ and

$$\int_0^1 (const)d\mu = 1 . \qquad (6)$$

This simply states that the probability that a scattered neutron emerge into some allowed direction is unity. A consequence of this normalization is that

$$f(\mu'\rightarrow\mu) = 1 . \qquad (7)$$

Angular approximation is accomplished by discretizing equation (5) in the $N/2$ forward directions of the previously described quadrature set. This gives the S_N approximation

$$\mu_j \frac{d\psi_j(x)}{dx} + \sigma^t\psi_j(x) = \sigma^s \sum_{n=1}^{N/2} \omega_n\psi_n(x) + q_j(x) \qquad (8)$$

with $j = 1,2,\cdots,N/2$. Quadrature weights obey

$$\sum_{n=1}^{N/2} \omega_n = 1 \qquad (9)$$

and all $\mu_j > 0$. The angular quadrature set is the positive half of the one used for the full-range problem.

Spatial discretization is accomplished in the same manner described in the previous chapter to give

$$d_j\psi_{k+1,j} - e_j\psi_{k,j} = \frac{\sigma^s}{2} \sum_{n=1}^{N/2} \omega_n(\psi_{k+1,n}+\psi_{k,n}) + q_{k+1/2,j} \qquad (10a)$$

for $j = 1,2,\cdots,N/2$ and $k = 1,2,\cdots,R$

where

$$d_j = \frac{2\mu_j + \Delta\sigma^t}{2\Delta} \qquad (10b)$$

and

$$e_j = \frac{2\mu_j - \Delta\sigma^t}{2\Delta} \qquad (10c)$$

and R is the number of equi-spaced spatial increments, Δ.

The S_N algorithm, as before, solves this system of equations iteratively as described by

$$d_j\psi_{k+1,j}^{(i+1)} - e_j\psi_{k,j}^{(i+1)} = \frac{\sigma^s}{2} \sum_{n=1}^{N/2} \omega_n\left(\psi_{k+1,n}^{(i)}+\psi_{k,n}^{(i)}\right) + q_{k+1/2,j} \qquad (11)$$

where i is an iteration index. This set of equations is divided by d_j to give the computer-solved S_N algorithm

$$\psi_{k+1,j}^{(i+1)} = \frac{e_j}{d_j}\psi_{k,j}^{(i+1)} + \frac{1}{d_j}\left\{\frac{\sigma_s}{2}\sum_{n=1}^{N/2}\omega_n\left(\psi_{k+1,n}^{(i)}+\psi_{k,n}^{(i)}\right) + q_{k+1/2,j}\right\}. \tag{12}$$

Numerical results for this algorithm with the first collision source

$$q_{k+1/2,j} = e^{-\left(\sigma_t x_{k+1/2}\right)/\mu_j} \tag{13}$$

and vacuum boundary conditions

$$\psi_{1,j} = 0 \tag{14}$$

for all j, will be displayed and discussed later.

Iterative Matrix Formulation

The S_N algorithm (12), (14) with source (13) has the matrix description

$$(D-E)\underline{\psi}^{(i+1)} = S\underline{\psi}^{(i)} + \underline{q} \tag{15}$$

derived from equations (11).

If the equations are arranged so that the solution vector has the order

$$\underline{\psi} = \begin{bmatrix} \psi_{2,1} \\ \psi_{3,1} \\ \vdots \\ \psi_{R+1,1} \\ \psi_{2,2} \\ \psi_{3,2} \\ \vdots \\ \psi_{R+1,2} \\ \vdots \\ \vdots \\ \psi_{R+1,N/2} \end{bmatrix} \tag{16}$$

then

$$\underline{q} = \begin{bmatrix} q_{3/2,1} \\ q_{5/2,1} \\ \vdots \\ q_{\frac{R+1}{2},1} \\ q_{3/2,2} \\ q_{5/2,2} \\ \vdots \\ q_{\frac{R+1}{2},2} \\ \vdots \\ \vdots \\ q_{\frac{R+1}{2},N/2} \end{bmatrix} . \tag{17}$$

The matrix (D-E) has the block form

$$\begin{bmatrix} \boxed{B_1} & & & \bigcirc \\ & \boxed{B_2} & & \\ & & \ddots & \\ \bigcirc & & & \boxed{B_{N/2}} \end{bmatrix} \tag{18}$$

where each block has the lower bi-diagonal form

$$\begin{bmatrix} d_i & & & & \bigcirc \\ -e_i & d_i & & & \\ & -e_i & d_i & & \\ & & \ddots & \ddots & \\ \bigcirc & & & -e_i & d_i \end{bmatrix} = B_i \ . \tag{19}$$

The index i is that for the ordinate μ_i. Each block is an R by R submatrix.

The scattering matrix has the form

$$\begin{array}{|c|c|c|c|} \hline S_{BD_1} & S_{BD_2} & \cdots & S_{BD_{N/2}} \\ \hline S_{BD_1} & S_{BD_2} & \cdots & S_{BD_{N/2}} \\ \hline \vdots & \vdots & & \vdots \\ \hline S_{BD_1} & S_{BD_2} & \cdots & S_{BD_{N/2}} \\ \hline \end{array} = S \tag{20}$$

where each block has the lower bi-diagonal form,

$$\begin{bmatrix} s_j & & & & \bigcirc \\ s_j & s_j & & & \\ & s_j & s_j & & \\ & & \ddots & \ddots & \\ \bigcirc & & & s_j & s_j \end{bmatrix} = S_{BD_j} \tag{21}$$

and

$$s_j = \frac{\sigma^s}{2}\,\omega_j \,. \tag{22}$$

Observe that this matrix formulation is the upper left quadrant of blocks of the full-range S_N matrix formulation and only the elements s_j are different.

Due to the block diagonal structure of (D-E), and the bi-diagonal form of each block, $(D-E)^{-1}$ is simply

$$\begin{bmatrix} B_1^{-1} & & & \bigcirc \\ & B_2^{-1} & & \\ & & \ddots & \\ \bigcirc & & & B_{N/2}^{-1} \end{bmatrix} = (D-E)^{-1} \tag{23}$$

where

$$
\begin{array}{c}
\\ \\ \\ \\ \\ k^{th}\ row \\ \\ \\ \\ \\ \\
\end{array}
\begin{bmatrix}
\frac{1}{d_i} & & & & & & \\
\frac{e_i}{d_i^2} & \frac{1}{d_i} & & & & & \\
\frac{e_i^2}{d_i^3} & \frac{e_i}{d_i^2} & \frac{1}{d_i} & & & & \\
\vdots & & & \ddots & & & \\
\frac{e_i^{k-1}}{d_i^k} & \frac{e_i^{k-2}}{d_i^{k-1}} & \cdots & & \ddots & & \\
 & & & & \frac{1}{d_i} & & \\
\vdots & \vdots & & & & & \\
\frac{e_i^{R-2}}{d_i^{R-1}} & \frac{e_i^{R-3}}{d_i^{R-2}} & \cdots & & \frac{e_i}{d_i^2} & \frac{1}{d_i} & \\
\frac{e_i^{R-1}}{d_i^{R}} & \frac{e_i^{R-2}}{d_i^{R-1}} & \cdots & & \frac{e_i^2}{d_i^3} & \frac{e_i}{d_i^2} & \frac{1}{d_i}
\end{bmatrix} = B_i^{-1} . \tag{24}
$$

and i is the index for μ_i. Each block is of dimension R by R. $(D-E)^{-1}$ can be applied in its general form to equation (15) to produce

$$
\underline{\psi}^{(i+1)} = (D-E)^{-1} S \underline{\psi}^{(i)} + (D-E)^{-1} \underline{q} \ . \tag{25}
$$

In this equation

$$
(D-E)^{-1} S = \begin{array}{|c|c|c|c|}
\hline
B_1^{-1} S_{BD_1} & B_1^{-1} S_{BD_2} & \cdots\cdots & B_1^{-1} S_{BD_{N/2}} \\
\hline
B_2^{-1} S_{BD_1} & B_2^{-1} S_{BD_2} & \cdots\cdots & B_2^{-1} S_{BD_{N/2}} \\
\hline
\vdots & \vdots & & \vdots \\
\hline
B_{N/2}^{-1} S_{BD_1} & B_{N/2}^{-1} S_{BD_2} & \cdots\cdots & B_{N/2}^{-1} S_{BD_{N/2}} \\
\hline
\end{array} \ . \tag{26}
$$

Observe that this matrix is the upper-left quadrant of blocks of the full-range iteration matrix of the previous chapter.

Each block has the form

$$
\begin{array}{r}
\\ \\ \\ \\ k^{th}\ \text{row} \\ B_i^{-1}S_{BD_j} = \omega_j \frac{\sigma^s}{2} \\ \\ \\ \\
\end{array}
\left[\begin{array}{c|c|c|c|c|c|c}
\frac{1}{d_i} & & & & & & \\
\hline
\frac{e_i}{d_i^2}+\frac{1}{d_i} & \frac{1}{d_i} & & & & & \\
\hline
\frac{e_i^2}{d_i^3}+\frac{e_i}{d_i^2} & \frac{e_i}{d_i^2}+\frac{1}{d_i} & \frac{1}{d_i} & & & \bigcirc & \\
\vdots & & & \ddots & & & \\
\frac{e_i^{k-1}}{d_i^k}+\frac{e_i^{k-2}}{d_i^{k-1}} & \frac{e_i^{k-2}}{d_i^{k-1}}+\frac{e_i^{k-3}}{d_i^{k-2}} & \frac{e_i^{k-3}}{d_i^{k-2}}+\frac{e_i^{k-4}}{d_i^{k-3}} & \cdots & & & \\
\vdots & \vdots & \vdots & & & & \\
\hline
\frac{e_i^{R-3}}{d_i^{R-2}}+\frac{e_i^{R-4}}{d_i^{R-3}} & \frac{e_i^{R-4}}{d_i^{R-3}}+\frac{e_i^{R-5}}{d_i^{R-4}} & \frac{e_i^{R-5}}{d_i^{R-4}}+\frac{e_i^{R-6}}{d_i^{R-5}} & & \frac{1}{d_i} & & \\
\hline
\frac{e_i^{R-2}}{d_i^{R-1}}+\frac{e_i^{R-3}}{d_i^{R-2}} & \frac{e_i^{R-3}}{d_i^{R-2}}+\frac{e_i^{R-4}}{d_i^{R-3}} & \frac{e_i^{R-4}}{d_i^{R-3}}+\frac{e_i^{R-5}}{d_i^{R-4}} & \cdots\cdots & \frac{e_i}{d_i^2}+\frac{1}{d_i} & \frac{1}{d_i} & \\
\hline
\frac{e_i^{R-1}}{d_i^{R}}+\frac{e_i^{R-2}}{d_i^{R-1}} & \frac{e_i^{R-2}}{d_i^{R-1}}+\frac{e_i^{R-3}}{d_i^{R-2}} & \frac{e_i^{R-3}}{d_i^{R-2}}+\frac{e_i^{R-4}}{d_i^{R-3}} & \cdots\cdots & \frac{e_i^2}{d_i^3}+\frac{e_i}{d_i^2} & \frac{e_i}{d_i^2}+\frac{1}{d_i} & \frac{1}{d_i}
\end{array}\right]
\quad (27)
$$

Calculation of the Norm

The infinity-norm of $(D-E)^{-1}S$, as previously defined, is easily calculated. The block row sum is

$$B_i^{-1}S_{BD_1}+B_i^{-1}S_{BD_2}+\cdots+B_i^{-1}S_{BD_{N/2}} = \frac{\sigma^s}{2}(\omega_1+\omega_2+\cdots+\omega_{N/2})(B_i^{-1}S_{BD})$$

$$= \frac{\sigma^s}{2}\, B_i^{-1}S_{BD} \qquad (28)$$

due to (9). Here S_{BD} has the form of (21) but with unity in each non-zero matrix element. Actually (28) is $\frac{\sigma^s}{2}$ times

the matrix part of equation (27). Consequently

$$||(D-E)^{-1}S|| = \frac{\sigma^s}{2}\left[\frac{|e_i|^{R-1}}{d_i^R} + 2\left(\frac{1}{d_i}+\frac{|e_i|}{d_i^2}+\frac{|e_i|^2}{d_i^3}+\cdots+\frac{|e_i|^{R-2}}{d_i^{R-1}}\right)\right] . \tag{29}$$

where as before, i is an index for the minimum absolute ordinate value in the set $\{\mu_j\}$. If one defines a β

$$0 \leq \frac{|e_i|}{d_i} = \beta < 1 \tag{30}$$

equation (29) becomes

$$||(D-E)^{-1}S|| = \frac{\sigma^s}{2d_i}\left[\, 2\left(\frac{1-\beta^R}{1-\beta}\right) - \beta^{R-1}\,\right] . \tag{31}$$

Sufficiency conditions for $||(D-E)^{-1}S|| < 1$ are derived by requiring that

$$\frac{\sigma^s}{d_i}\,\frac{1}{1-\beta} < 1$$

or that

$$2(\mu_i)_{min} + \Delta\sigma^t > 2\Delta\sigma^s + |2(\mu_i)_{min} - \Delta\sigma^t| . \tag{32}$$

This is the same inequality which gave the sufficiency conditions in Chapter II for convergence of the full-range S_N algorithm to the exact solution $\underline{\psi}^*$. These conditions are the same for both problems and will not be restated here.

Exponential Spatial Transform

The transform

$$\psi_j(x) = \phi_j(x)e^{\alpha x} , \tag{33}$$

where α is a constant greater than zero, applied to the angularly discretized equations (8) gives

$$\mu_j \frac{d\phi_j(x)}{dx} + (\sigma^t + \alpha\mu_j)\phi_j(x) = \sigma^s \sum_{n=1}^{N/2} \omega_n \phi_n(x) + q_j(x)e^{-\alpha x} \tag{34}$$

$j = 1,2,\cdots,N/2$. Observe that the transform has added absorption to the problem in amount $\alpha\mu_j$ to produce a new "effective" total cross section, $(\sigma^t + \alpha\mu_j)$ as well as changed the source.

Spatially discretizing these equations in the same way as previously described for the unmodified equations gives

$$d_j\phi_{k+1,j}^{(i+1)} - e_j\phi_{k,j}^{(i+1)} = \frac{\sigma^s}{2} \sum_{n=1}^{N/2} \omega_n \left(\phi_{k+1,n}^{(i)} + \phi_{k,n}^{(i)}\right) + q_{k+\frac{1}{2},j}e^{-\alpha x_{k+\frac{1}{2}}} \tag{35}$$

$k = 1,2,\cdots,R$ and $j = 1,2,\cdots,N/2$. The algorithm in machine form is

$$\phi_{k+1,j}^{(i+1)} = \frac{e_j}{d_j}\phi_{k,j}^{(i+1)} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N/2}\omega_n\left(\phi_{k+1,n}^{(i)} + \phi_{k,n}^{(i)}\right) + q_{k+\frac{1}{2},j}e^{-\alpha x_{k+1/2}}\right\} \tag{36}$$

where

$$d_j = d_j + \frac{\alpha\mu_j}{2} = \frac{2\mu_j + \Delta\sigma^t + \Delta\alpha\mu_j}{2\Delta} \tag{37a}$$

and

$$e_j = e_j - \frac{\alpha\mu_j}{2} = \frac{2\mu_j - \Delta\sigma^t - \Delta\alpha\mu_j}{2\Delta} . \qquad (37b)$$

Matrix Formulation

The ϕ-domain algorithm has precisely the same matrix form as the unmodified S_N equations. That is, equations (11) through (27) describe the ϕ-domain algorithm if e_j and d_j are replaced by e_j and d_j and if source and solution vectors are changed accordingly.

The infinity norm of the transform domain iteration matrix is

$$||(D-E)^{-1}S|| = \frac{\sigma^s}{2}\left[\frac{|e_i|^{R-1}}{d_i^R} + 2\left(\frac{1}{d_i} + \frac{|e_i|}{d_i^2} + \frac{|e_i|^2}{d_i^3} + \cdots + \frac{|e_i|^{R-2}}{d_i^{R-1}}\right)\right] \qquad (38)$$

where i is the same index as before. Defining

$$0 \le \frac{|e_i|}{d_i} = \gamma < 1 \qquad (39)$$

The above equation becomes

$$||(D-E)^{-1}S|| = \frac{\sigma^s}{2d_i}\left[2\left(\frac{1-\gamma^R}{1-\gamma}\right) - \gamma^{R-1}\right] . \qquad (40)$$

Properties of $||(D-E)^{-1}S||$

The infinity norm of the transform domain iteration matrix is a monotone decreasing function of the acceleration parameter, α. For small α, the slope of $||(D-E)^{-1}S||$ with respect to α is steep but decreases to near zero at $\alpha = \alpha^*$ where

$$\alpha^* = \frac{2(\mu_i)_{min} - \Delta\sigma^t}{\Delta(\mu_i)_{min}} \tag{41}$$

which is the value of α when e_i is zero. The norm value at α^* is

$$||(D-E)^{-1}S||_{\alpha=\alpha^*} = \frac{\Delta\sigma^s}{2(\mu_i)_{min}} \; . \tag{42}$$

From this point the norm slowly approaches zero as α approaches ∞. This behavior of $||(D-E)^{-1}S||$ with α is verified by analysis of equation (40) in Appendix B.

Optimum Value of α

Appendix B reveals the behavior of $||(D-E)^{-1}S||$ for $\alpha > \alpha^*$ to be

$$||(D-E)^{-1}S|| = \frac{\Delta\sigma^s}{2(\mu_i)_{min}}\left\{1 - \frac{(\Delta\varepsilon)^R + 2(\Delta\varepsilon)^{R-1}}{(4 + \Delta\varepsilon)^R}\right\} \tag{43}$$

where

$$\alpha = \alpha^* + \varepsilon \tag{44}$$

and

$$0 < \varepsilon < \infty \; . \tag{45}$$

In the region where $\Delta\varepsilon < 1$, $||(D-E)^{-1}S||$ essentially takes its value at α^*. Consequently very little norm reduction occurs in this region. This defines a practical "optimum" value for α, that is, $\alpha = \alpha^*$. Figure 3.1 illustrates this "optimum" value for a particular problem. Observe that the

THE INFINITY NORM OF THE HALF-RANGE DISCRETE ORDINATES UNMODIFIED ITERATION MATRIX IS COMPARED TO THAT OF THE TRANSFORM DOMAIN ALGORITHM FOR VARIOUS VALUES OF THE ACCELERATION PARAMETER, ALPHA

THE PROBLEM PARAMETERS ARE:
SLAB WIDTH= 10.0 CM.
N= 30 SPATIAL INTERVALS.
DELTA=0.3333 CM.
SIGMA=1.00000 1/CM.
SIGMAS=0.99990 1/CM.
ANGULAR QUADRATURE SET=SIX POINT GAUSS.
MINIMUM ABSOLUTE MU(I)=0.23862
UNMODIFIED ALGORITHM COEFFICIENT E(1)/D(1)= 0.17754 .
UNMODIFIED ALGORITHM COEFFICIENT 1/D(1)= 0.82246 .
UNMODIFIED ALGORITHM NORM= 0.99990 .

ALPHA	E(1)/D(1)	1/D(1)	INFINITY NORM OF ITERATION MATRIX
0.20	0.15487	0.80663	0.95435
0.40	0.13306	0.79140	0.91278
0.60	0.11206	0.77673	0.87467
0.80	0.09183	0.76260	0.83962
1.00	0.07231	0.74897	0.80727
1.20	0.05349	0.73582	0.77732
1.40	0.03531	0.72312	0.74951
1.60	0.01775	0.71086	0.72363
1.80	0.00077	0.69900	0.69947
2.00	-0.01565	0.68753	0.69839
2.20	-0.03154	0.67643	0.69839
2.40	-0.04692	0.66569	0.69839
2.60	-0.06182	0.65528	0.69839
2.80	-0.07627	0.64519	0.69839
3.00	-0.09027	0.63541	0.69839
3.20	-0.10386	0.62592	0.69839
3.40	-0.11705	0.61671	0.69839
3.60	-0.12985	0.60776	0.69839
3.80	-0.14229	0.59908	0.69839
4.00	-0.15438	0.59063	0.69839
4.20	-0.16613	0.58242	0.69839
4.40	-0.17756	0.57444	0.69839
4.60	-0.18868	0.56667	0.69839
4.80	-0.19951	0.55911	0.69839
5.00	-0.21005	0.55175	0.69839

Figure 3.1 Iteration Matrix Norm Behavior with α.

norm is not further reduced for values of $\alpha > 2.00$, at least to five significant figures. For this problem, $\alpha^* = 1.80924$. Hence an S_N problem of the type discussed here has a practical "optimum" acceleration parameter which can quickly be calculated from equation (41). Use of this value will result in the largest practical norm reduction with accompanying improved iteration efficiency.

Range of Utility of Transform Method

From the previous discussion, it is obvious that $||(D-E)^{-1}S||$ has reached a practical minimum value when e_i' is zero. Consequently the transform method can significantly decrease the norm of the unmodified S_N iteration matrix only when $(D-E)^{-1}S$ is non-negative, that is, when

$$\Delta < \frac{2(\mu_i)_{min}}{\sigma^t} \quad . \qquad (46)$$

Otherwise e_i is negative, which guarantees that e_i is negative for all $\alpha \geq 0$. In this case, the norm has already reached a value which decreases very slowly with increased α.

The transform method has the greatest latitude for norm reduction, if (46) is met, when Δ is chosen very small. Equation (41) reveals that α^* is large when Δ is chosen small. A large α^* decreases the norm of $||(D-E)^{-1}S||$ more than a small α^*. Table V shows this property for a specific problem.

TABLE V

α^* FOR VARIOUS Δ

R	Δ	α^*	$\|\|(D-E)^{-1}S\|\|$ at α^*
30	0.333	1.81	0.69839
50	0.200	5.81	0.41903
100	0.100	15.81	0.20952

Problem parameters:

slab width = 10.0 cm.

$\sigma^t = 1.0\ \text{cm}^{-1}$.

$\sigma^s = 0.9999\ \text{cm}^{-1}$.

$(\mu_i)_{min} = 0.23862$

$||(D-E)^{-1}S|| = 0.9999$.

Table V shows that small Δ allows large values of α to be used resulting in vast reductions in $||(D-E)^{-1}S||$ and improved convergence efficiency. The choice of α, however is based primarily on the estimation of tolerable discretization error for a given problem. Consequently the transform method's utility depends upon factors other than those which most improve iterative efficiency. Once a Δ is chosen, however, α^* is easily determined. Its use provides practical maximum acceleration of convergence for that problem.

Computer Experiment

The half-range problem algorithm previously described in this chapter was run on an IBM 360/67 computer with the following problem parameters:

Slab width = 10.0 cm

R = 30 spatial intervals

$\Delta = 0.3333$ cm

$\sigma^t = 1.0\ \text{cm}^{-1}$

$\sigma^{s} = 0.9999 \text{ cm}^{-1}$

Six ordinate Gauss quadrature set

First collision source of unit strength

Fractional iterative error less than $\varepsilon = 10^{-4}$.

The improved convergence criterion, described in Chapter II, was used. A program listing and sample output for unmodified S_N are included in Appendix C. The transform method program and sample output are included in Appendix D.

Table VI compares the number of iterations to convergence for various values of the acceleration parameter, α.

TABLE VI

TRANSFORM METHOD RESULTS

Alpha	Infinity Norm	Number of Iterations to Converge	Percent Decrease in Number of Iterations	Maximum Percentage Difference in Unmodified to Transform Solutions
0	0.99990	59	--	0
0.2	0.95435	48	18.7	0.36
0.5	0.89332	46	22.0	1.1
1.0	0.80727	45	23.8	7.1
1.809*	0.69839	45	23.8	38.9
2.0	0.69839	45	23.8	49.4

Observe that $\alpha = 0$ is the unmodified S_N result. The symbol * signifies the optimum α. For $R = 30$, the condition

$\Delta < \frac{2(\mu_i)_{min}}{\sigma^t}$ is met and the unmodified S_N iteration matrix is non-negative.

Table VI reveals that the transform method accelerates convergence of the S_N algorithm for the half-range problem. The acceleration parameter, α, decreases e_j and increases d_j so that each non-zero element of the iteration matrix is decreased in value (see equations (26) and (27)). This produces a reduced norm and concomitant acceleration of convergence.

Values of α greater than α^* used here do not further reduce the norm or number of iterations to convergence. The dependence of convergence efficiency on the norm is not linear. Acceptable improvement of convergence efficiency is attained for values of α well below α^*.

Calculational Effort

The calculational effort of a machine algorithm is based upon the total number of machine operations performed. Since additions and subtractions require only an infinitesimal amount of computer time compared to multiplications and divisions, only the latter are tabulated. Table VII lists the number of operations in each component part of the unmodified S_N and transform method algorithms. Here n is equal to $(R \cdot N/2)$, i is the number of iterations to convergence of the unmodified algorithm and j is the number of iterations required for transform algorithm convergence.

TABLE VII

COMPUTATIONAL EFFORT COMPARISON

Quantity	Unmodified S_N Algorithm	Transform Method Algorithm
Mesh sweep	$i\cdot(2n)$	$j\cdot(2n)$
Scattering source	$i[N/2+2]$	$j[N/2+2]$
Coefficients e, d	$9(N/2)$	$13(N/2)$
Transform source	0	n
Transform solution	0	n
Total	$i[2n+\frac{N}{2}+2]+9(\frac{N}{2})$	$j[2n+\frac{N}{2}+2]+2n+13(\frac{N}{2})$

The transform method reduces computational work if,

$$j < i - 1 - \frac{4(3N-1)}{4R+3N+2} \, . \tag{47}$$

For most problems $R >> N$ so the transform method has utility even if only a few iterations are saved in its employment.

Isaacson[53] shows that Gaussian elimination requires

$$\frac{n(n^2-1)}{3} + n^2 \tag{48}$$

operations to solve a matrix system corresponding to the exact solution, $\underline{\psi}^*$. Comparison of the S_N iterative technique shows a significant economy for large systems, n large, if i is not too large. Even so, i would have to be of the order n^2 before the computational effort of the unmodified S_N algorithm reached that of Gauss elimination.

Error in Transform Method Solution

The new convergence criterion

$$\max_{j} \frac{\left[\phi_j^{(i+1)} - \phi_j^{(i)}\right]}{\phi_k^{(i)}} < \frac{\varepsilon}{1-\varepsilon}\left[1-\|(D-E)^{-1}S\|\right] \tag{49}$$

which guarantees that

$$\frac{\phi_k^* - \phi_k^{(i)}}{\phi_k^*} < \varepsilon \tag{50}$$

for all k, was used in the computer algorithm. Using the transform (33), inequality (50) implies that

$$\frac{\psi_k^* e^{-\alpha x_k} - \psi_k^{(i)} e^{-\alpha x_k}}{\psi_k^* e^{-\alpha x_k}} < \varepsilon$$

or that

$$\frac{\psi_k^* - \psi_k^{(i)}}{\psi_k^*} < \varepsilon \tag{51}$$

for all k, where $\underline{\psi}^{(i)}$ is the ψ-domain transform method final solution. Consequently the convergence criterion (49) and convergence properties of the method guarantee that the transform method ψ-domain solution has fractional iteration errors less than ε. This same criterion was met for the unmodified S_N algorithm. The origin of the discrepancy in the ψ-domain solutions for transform method and unmodified

S_N solutions, exhibited in column 5 of Table VI, is therefore not attributable to iteration error.

The origin of the discrepancy mentioned above is discretization error. The ϕ-domain exact solution, $\underline{\phi}^*$, has much greater variation over the slab width than does the unmodified S_N exact solution, $\underline{\psi}^*$. Figures 3.2 and 3.3 demonstrate this for a particular direction. Since the spatial discretization of the derivative is accurate to order Δ^2 in both domains, it is a poorer approximation in the ϕ-domain, where the solution has large curvature magnitude, than it is in the unmodified S_N ψ-domain where the solution has a relatively small variation over the slab width. Consequently larger discretization errors occur in the transform method ϕ-domain solution when large values of α are used. At α^*, $\underline{\phi}^{(i)}$ varied seven orders of magnitude over the slab width whereas the unmodified S_N ψ-domain solution varied by less than a factor of two. This accounts for large discretization errors in the ψ-domain transform method solution when α is near α^*. Table VI reveals, however, that as α is decreased from α^*, the discretization error diminishes so that the ψ-domain solutions for both algorithms agree to within less than 0.4% at $\alpha = 0.2$. For this α, the ϕ-domain solution varied less than a factor of five over the slab width, producing smaller discretization error than did larger α. Here convergence was accelerated to 78.5% of the maximum acceleration obtained at α^*. The value of $\alpha = 0.2$ was a good practical choice for this

FIGURE 3.2

SAMPLE UNMODIFIED S_N ψ-DOMAIN SOLUTION

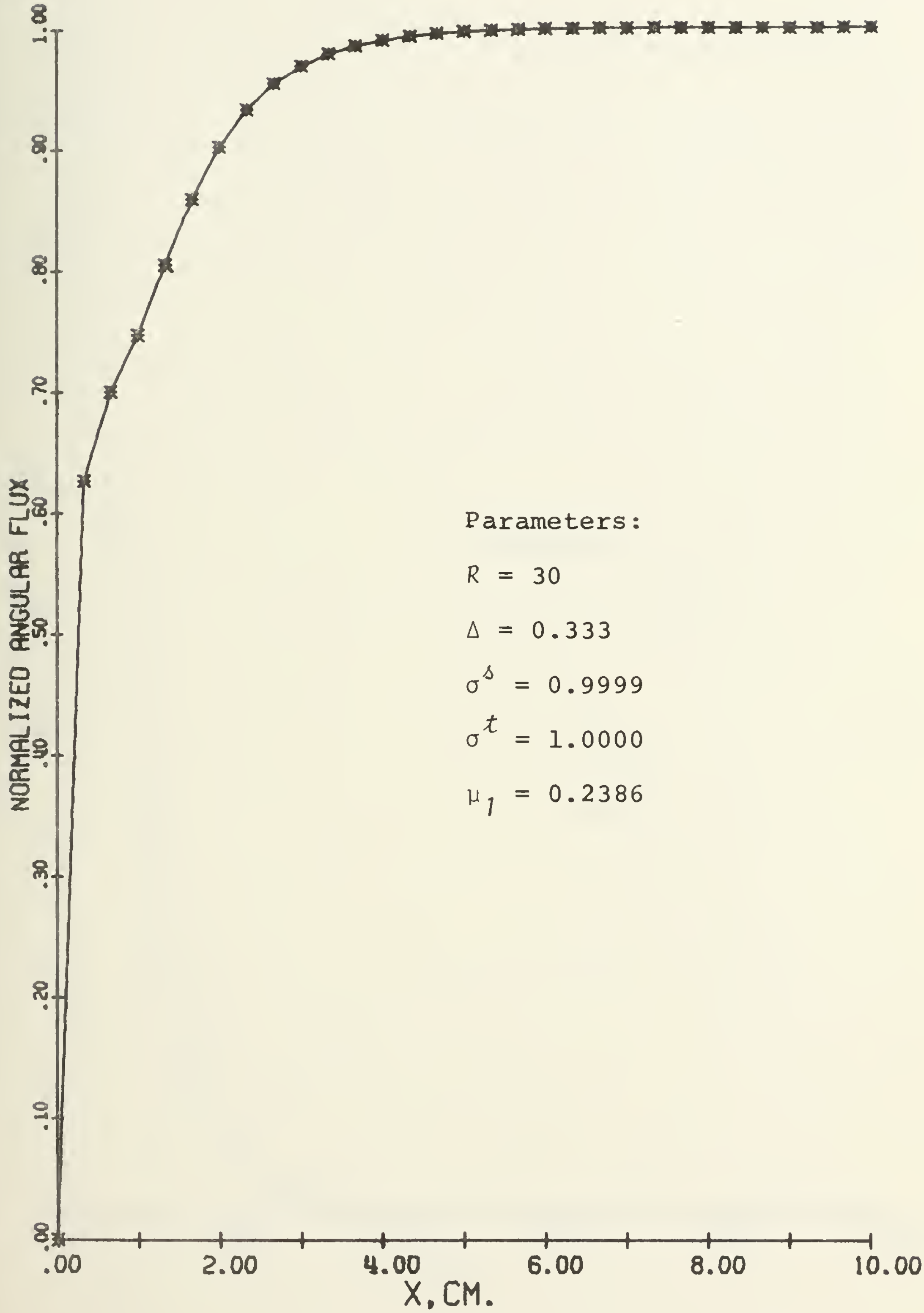
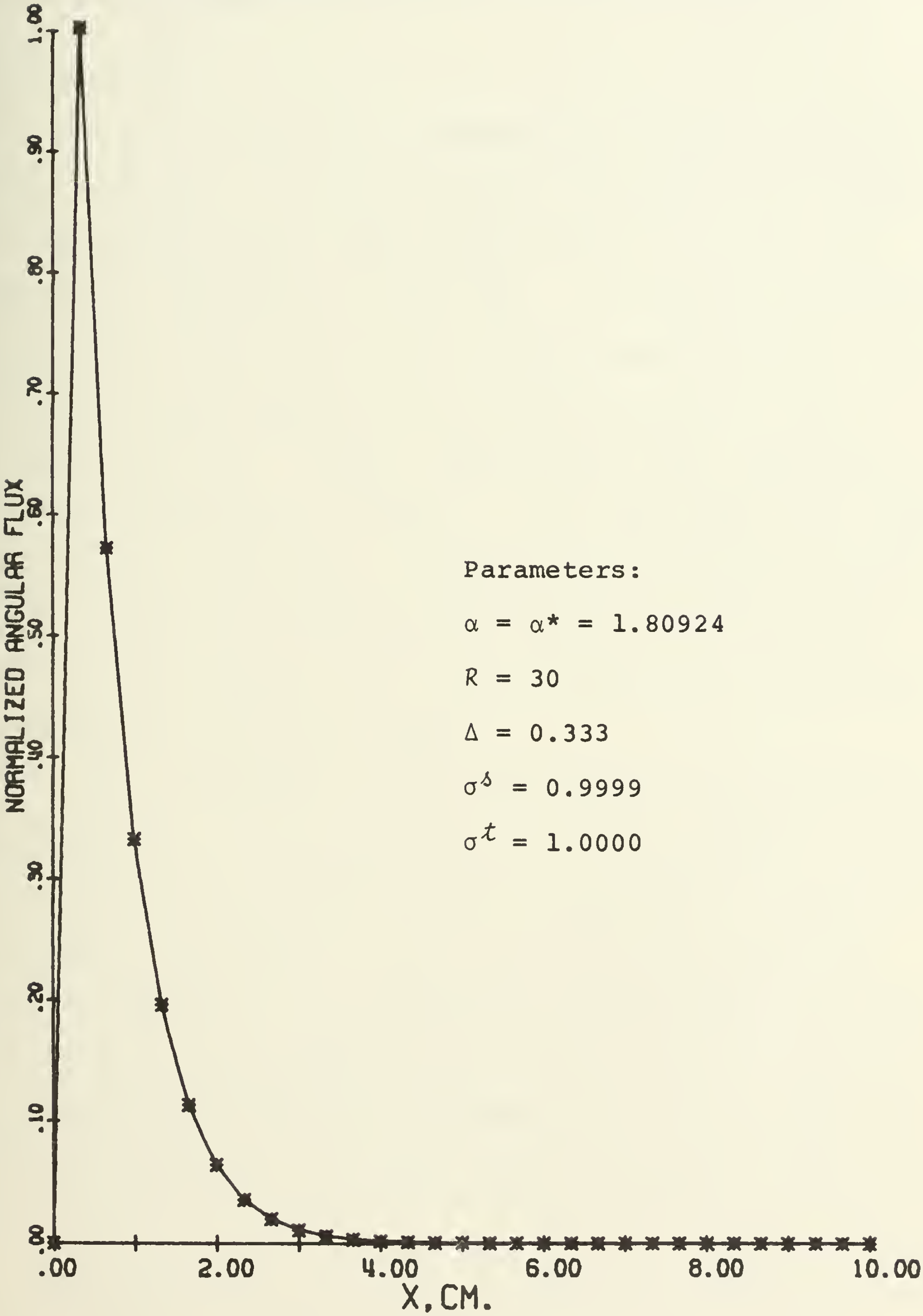


FIGURE 3.3

SAMPLE TRANSFORM METHOD ϕ-DOMAIN SOLUTION



problem. For other problems, the user must make a choice of α based on practical experience and a guess at the expected solution.

Influence of Δ

Previous discussion connected with Table V revealed that, in theory, a smaller Δ allows the choice of a larger α^* with accompanying larger decrease in $||(D-E)^{-1}S||$ and number of iterations to convergence. In practice, however, such large values of α are not useful because of the extremely large resulting discretization errors in the ϕ-domain solution. Another consideration also limits the practical usefulness of large α. For thick slabs, the transformed source, which has a factor $e^{-\alpha x}$, can become extremely small. Since all computers have a limit on the smallest number it can handle, 16^{-65} for the IBM 360/67, it is easy to exceed this limit for large α.

As a consequence of the above factors, it is best to use values of α much smaller than α^*, for a problem with any Δ for which the transform method is applicable. Computer experiments conducted by this investigator have verified that the choice of $\alpha = 0.2$ produces approximately the same acceleration of convergence and accuracy listed in Table VI for the problem described previously but with the much smaller values of Δ resulting from $R = 50$ and $R = 100$.

Generalization of Transform Method

The transform method immediately generalizes for the half-range problem treated here to include heterogeneous slabs. In this case the cross sections are functions of position. The relatively simple expressions for $||(D-E)^{-1}S||$ and $||(\mathcal{D}-\mathcal{E})^{-1}\mathcal{S}||$ shown in this chapter do not occur but the matrix block form prevails and these norms can be calculated.

Transforms such as equations (3) and (4) place space dependent absorption into the problem. When using transforms of this type, care must be exercised to restrict the range of the arguments in order that finite non-negative absorption is added to the problem by the transform. Otherwise reduced or negative "effective" total cross sections will result and convergence will be decelerated in the transform domain.

It is possible that problems in r,μ and higher-dimensional geometries can be formulated in the mathematical context displayed here for the simplest geometry. If transforms can be found which place an acceleration parameter in the proper place in the iteration matrix to reduce its norm, vast savings of computational effort can be expected for these problems. The transform must be one which leaves the angular discretized S_N equations invariant. To be of practical value, such a transform must not only properly position an acceleration parameter in the iteration matrix but also must not alter the source so drastically that large

variations occur in the ϕ-domain solution with the accompanying unacceptable discretization error.

Possible transform candidates are some form of the integrating factor for the angular discretized equations describing the problem after neglecting the scattering gain term. Transforms are not restricted to this class, however, and will prove useful if they meet the criterion described above.

It is doubtful that a transform can be found which is useful for all problems in a particular geometry. Experience of this investigator indicates that it is more likely that each problem in the geometry has a "best" transform which can be prescribed based upon the source, boundary conditions, and the prescriber's estimate of the form of the solution.

Conclusions

1) The transform (33) applied to the half-range slab problem accelerates convergence of the S_N algorithm for predominantly scattering media using the improved convergence criterion.

2) The transform method is successful only for problems in which Δ is chosen so that $\Delta < \frac{2(\mu_i)_{min}}{\sigma^t}$.

3) Significant convergence acceleration occurs for values of α much smaller than α^*. These smaller values of α produce solutions with acceptable discretization errors.

4) A practical choice of α for a particular problem must be based upon experience and the user's estimate of the expected solution characteristics.

CHAPTER IV

DISCRETIZATION ERROR IMPROVEMENT USING SPATIAL TRANSFORMS

Discretization error or truncation error, as it is frequently called, results from discretizing an equation in the continuous domain and comparing the discretized equation to the analytic equation. For the full-range S_N approximation of Chapter II, this process is illustrated by approximating the set of ordinary differential equations

$$\mu_j \frac{d\ _j(x)}{dx} + \sigma^t \psi_j(x) = \frac{\sigma^s}{2} \sum_{n=1}^{N} \omega_n \psi_n(x) + q_j(x) \tag{1}$$

$j = 1,2,\cdots,N$, which are valid in the spatial continuum $0 \leq x \leq L$, by the spatially discretized set

$$\mu_j\left(\frac{\psi_{k+1,j}-\psi_{k,j}}{\Delta}\right) + \sigma^t\left(\frac{\psi_{k+1,j}+\psi_{k,j}}{2}\right) = \frac{\sigma^s}{2} \sum_{n=1}^{N} \omega_n\left(\frac{\psi_{k+1,n}+\psi_{k,n}}{2}\right) + q_{k+1/2,j} \tag{2}$$

which uses a centered difference approximation for the derivative and a simple average for $\psi_{k+1/2,j}$ at each midpoint $X_{k+1/2} = \dfrac{X_{k+1}+X_k}{2}$.

Expanding $\psi_j(X_{k+1})$ and $\psi_j(X_k)$ in Taylor series about the point $\psi_j(X_{k+1/2})$, truncating appropriately, and forming the proper sums produces

$$\frac{\psi_j(X_{k+1})-\psi_j(X_k)}{\Delta} = \psi_j'(X_{k+1/2}) + \psi'''(\overline{X}_{k+1/2})\ O(\Delta^2) \tag{3}$$

where $X_k < \overline{X}_{k+1/2} < X_{k+1}$ (4)

and

$$\frac{\psi_j(X_{k+1})+\psi_j(X_k)}{2} = \psi_j(X_{k+1/2}) + \psi''(\overline{\overline{X}}_{k+1/2})\, O(\Delta^2) \qquad (5)$$

where $X_k < \overline{\overline{X}}_{k+1/2} < X_{k+1}$. (6)

Substituting equations (3) and (5) for the appropriately indexed quantities in equation (2) gives

$$\mu_j\left[\frac{d\psi_j(X_{k+1/2})}{dx} + \psi_j'''(\overline{X}_{k+\frac{1}{2}})O(\Delta^2)\right] + \sigma^t\left[\psi_j(X_{k+\frac{1}{2}}) + \psi_j''(\overline{X}_{k+\frac{1}{2}})O(\Delta^2)\right] - \frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\left[\psi_n(X_{k+\frac{1}{2}}) + \psi_n''(\overline{X}_{k+\frac{1}{2}})O(\Delta^2)\right] + \tau_j(X_{k+\frac{1}{2}}) \qquad (7)$$

Now evaluating $\psi_j(x)$ at $X_{k+1/2}$ in equations (1) and subtracting the result from equation (7) gives the discretization error

$$\tau_j(X_{k+1/2}) = O(\Delta^2)\psi_j'''(\overline{X}_{k+1/2}) + \sigma^t O(\Delta^2)\psi_j''(\overline{\overline{X}}_{k+1/2}) - \sigma^s O(\Delta^2)\sum_{n=1}^{N}\omega_n\psi_n''(\overline{\overline{X}}_{k+1/2}) . \qquad (8)$$

If it is assumed that the solution $\psi_j(X)$ to equation (1) is sufficiently smooth that its third derivative exists and is bounded, that is,

$$\psi_j'''(x) \leq M \qquad (9)$$

for all j,x, then

$$\tau_j(X_{k+1/2}) \leq MO(\Delta^2) + \rho O(\Delta^2) \qquad (10)$$

where

$$\rho = \max_i\left[\sigma^t\psi_i''(\overline{\overline{X}}_{k+1/2}) - \sigma^s\sum_{n=1}^{N}\omega_n\psi_n''(\overline{\overline{X}}_{k+1/2})\right] . \qquad (11)$$

The discretization error is clearly of order Δ^2 for this method. That is, τ approaches zero as Δ^2 approaches zero.

A standard technique for diminishing the discretization error is to choose Δ small enough, depending upon one's estimate of the variation of the solution, that the discretization error is tolerably small.

Transform Method

The above technique of diminishing Δ to improve the discretization error has a practical limit. As Δ decreases, the number of spatial mesh points increases as does the computational cost. Consequently a choice of Δ must be based on the tradeoff between accuracy and cost.

Another method of decreasing the discretization error is suggested by (10) and (11). If a transform can be found which renders equations (1) invariant in a transform domain and diminishes the second derivative of the solution without increasing M, the discretization error will be reduced. That is, if the transform domain solution of equation (1) has less curvature magnitude over the slab than the solution $\psi_j(x)$ of equation (1), the transform method will improve the discretization error. This method will be demonstrated in the next section for a simplified S_N problem which has an analytic solution.

Simplified Problem

The equation

$$\mu_0 \frac{d\psi(x)}{dx} + \sigma^t \psi(x) = \sigma^s \psi(x) + q(x) \tag{12}$$

with boundary condition

$$\psi(o) = 1.0 \tag{13}$$

and source

$$q(x) = e^{-\sigma^t x} \tag{14}$$

has a solution

$$\psi(x) = \frac{e^{-\sigma^t x}}{(\sigma^a - \mu_o \sigma^t)} + \frac{(\sigma^a - \mu_o \sigma^t - 1)}{(\sigma^a - \mu_o \sigma^t)} e^{-\sigma^a x/\mu_o} \tag{15}$$

where

$$\sigma^a = \sigma^t - \sigma^s \tag{16}$$

and μ_o is a positive real constant.

An S_N-type iterative solution of the same problem is formulated by approximating the derivative by a central difference and approximating equation (12) at a spatial midpoint as before to give

$$\psi_{k+1}^{(i+1)} = \frac{e}{d} \psi_k^{(i+1)} + \frac{1}{d}\left\{\sigma^s\left(\frac{\psi_{k+1}^{(i)} + \psi_k^{(i)}}{2}\right) + q_{k+1/2}\right\} \tag{17}$$

with

$$\psi_1 = 1.0 \tag{18}$$

and

$$q_{k+1/2} = e^{-\sigma^t x_{k+1/2}} \quad . \tag{19}$$

Here

$$e = \frac{2\mu_o - \Delta\sigma^t}{2\Delta} \tag{20a}$$

and

$$d = \frac{2\mu_o + \Delta\sigma^t}{2\Delta} \quad . \tag{20b}$$

Observe that this algorithm is precisely the same form as the S_N algorithm along a $|\mu_j|$ ray with the exception of the quadrature approximation of the scattering gain term.

The Exponential Transform

The transform

$$\psi(x) = \phi(x)\, e^{-\alpha x}, \tag{21}$$

with $\alpha > 0$, can be applied to equation (12) to give the transform domain equation

$$\mu_o \frac{d\phi(x)}{dx} + (\sigma^t - \alpha\mu_o)\phi(x) = \sigma^s\phi(x) + q(x)e^{\alpha x} \; . \tag{22}$$

Discretizing spatially, as before, gives

$$\phi_{k+1}^{(i+1)} = \frac{e}{d}\,\phi_k^{(i+1)} + \frac{1}{d}\left\{\sigma^s\left(\frac{\phi_{k+1}^{(i)} + \phi_k^{(i)}}{2}\right) + q_{k+1/2}\, e^{\alpha x_{k+1/2}}\right\} \tag{23}$$

where

$$e = \frac{2\mu_o - \Delta\sigma^t + \Delta\alpha\mu_o}{2\Delta} \tag{24a}$$

and

$$d = \frac{2\mu_o + \Delta\sigma^t - \Delta\alpha\mu_o}{2\Delta} \; . \tag{24b}$$

Observe that this transform places the parameter α into such a position that $\frac{e}{d}$ is increased and $\frac{1}{d}$ is increased. As shown in Chapter II, this increases the norm of the iteration matrix and decelerates convergence of the algorithm. Consequently the method is restricted to those problems for predominantly absorbing media in which the iterative efficiency is not so sensitive to the norm of the iteration matrix. For these problems, the conventional convergence criterion

$$\frac{\psi_k^{(i+1)} - \psi_k^{(i)}}{\psi_k^{(i)}} < \varepsilon \tag{25}$$

can be used with negligible loss in iterative precision.

The exact ϕ-domain solution of equation (22) with source

$$q(x)e^{\alpha x} = e^{-\sigma^t x+\alpha x} = e^{x(\alpha-\sigma^t)} \tag{26}$$

and boundary condition

$$\phi(o) = 1.0 \tag{27}$$

is $$\phi(x) = \frac{e^{-x(\sigma^t-\alpha)}}{(\sigma^a-\mu_o\sigma^t)} + \frac{(\sigma^a-\mu_o\sigma^t-1)}{(\sigma^a-\mu_o\sigma^t)}\, e^{-x\frac{(\sigma^a-\alpha\mu_o)}{\mu_o}} . \tag{28}$$

Sample Problem

A problem with the parameters listed below was treated by the methods described above.

Slab width = 10.0 cm.

R = 30 spatial intervals.

$\sigma^{t} = 1.0\ \text{cm}^{-1}$.

$\sigma^{s} = 0.2\ \text{cm}^{-1}$.

$\mu_0 = 0.9$.

$\varepsilon = 10^{-4}$.

The exact solution of equation (12) for these parameters is shown in Figure 4.1 and is

$$\psi(x) = 11e^{-.9x} - 10e^{-x} \ . \tag{29}$$

The exact ϕ-domain solutions for various values of α are shown in Figures 4.2 through 4.6. For values of $\alpha = 0.5$ through 0.9, the exact ϕ-domain solution has much less curvature magnitude over the slab width than does the exact ψ-domain solution. Consequently it is expected that the transform domain iterative algorithm (23) solution will possess smaller local discretization errors than the unmodified S_N algorithm (17).

Algorithms (17) and (23) were run on an IBM 360/67 computer to verify the expected discretization error improvement using the transform method. Table VIII summarizes the results.

The fractional errors were calculated from

$$\frac{|\psi_k^{exact} - \psi_k^{(i)}|}{\psi_k^{exact}} \tag{30}$$

FIGURE 4.1

EXACT SOLUTION FOR UNMODIFIED S_N

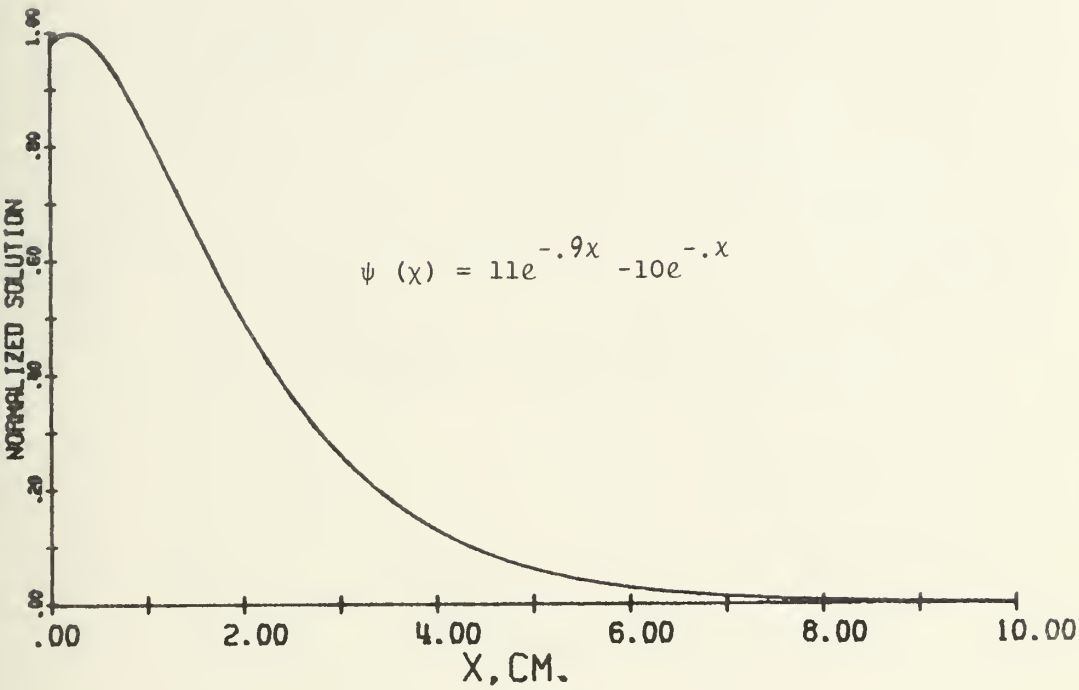


FIGURE 4.2

EXACT SOLUTION FOR α = 0.5

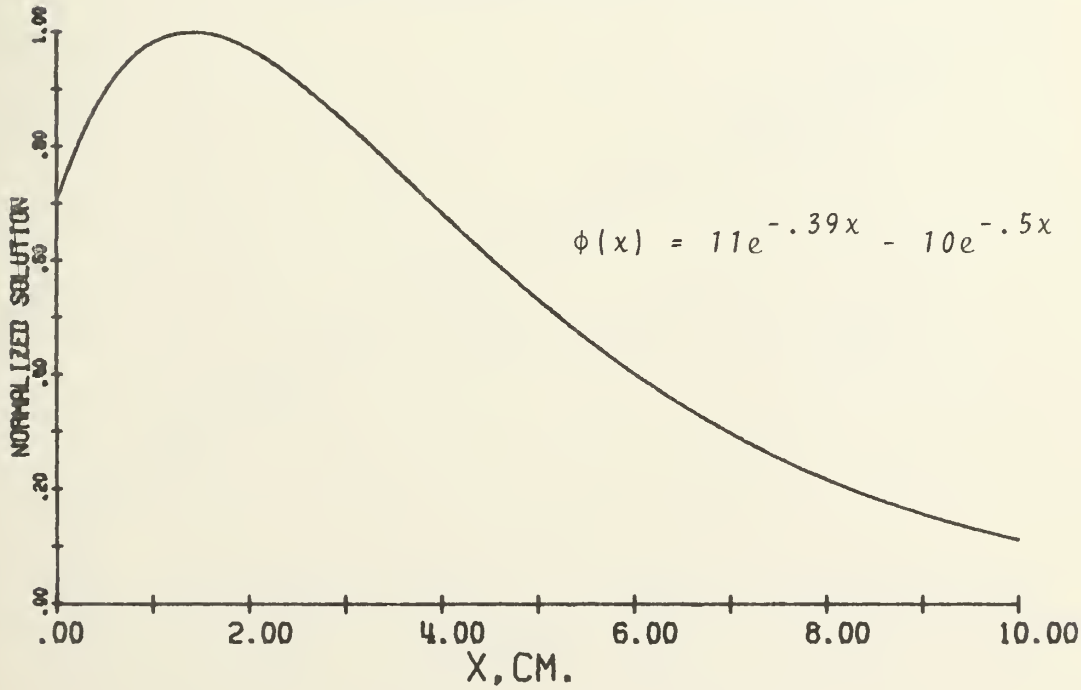


FIGURE 4.3

EXACT SOLUTION FOR α = 0.7.

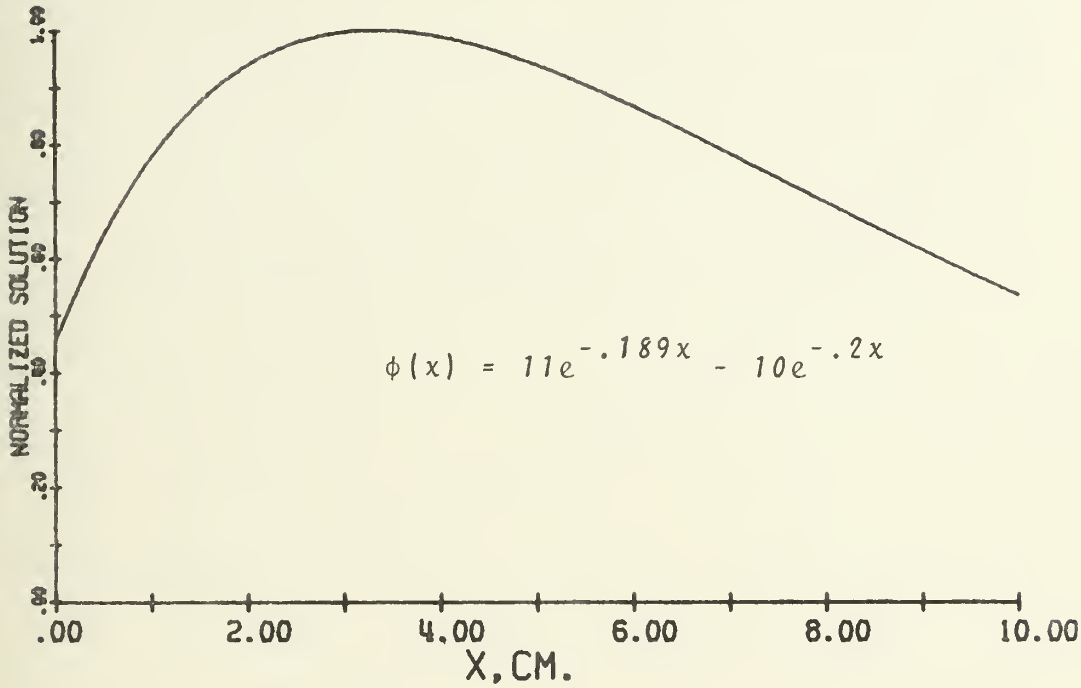


FIGURE 4.4

EXACT SOLUTION FOR α = 0.9

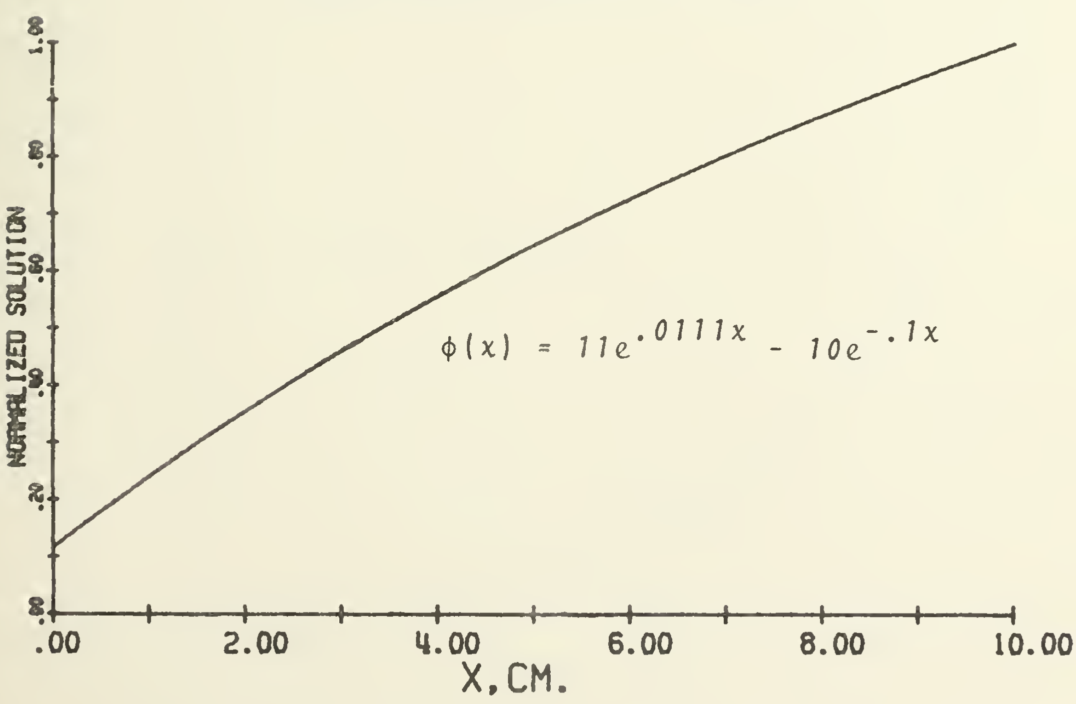


FIGURE 4.5

EXACT SOLUTION FOR α = 1.1

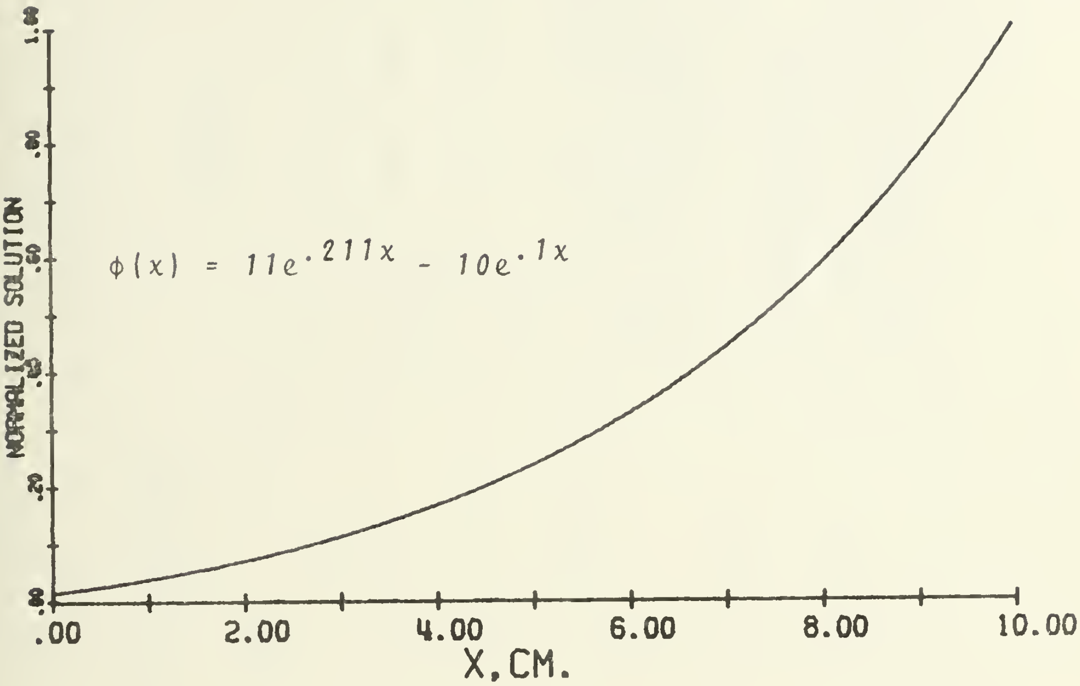


FIGURE 4.6

EXACT SOLUTION FOR α = 1.3

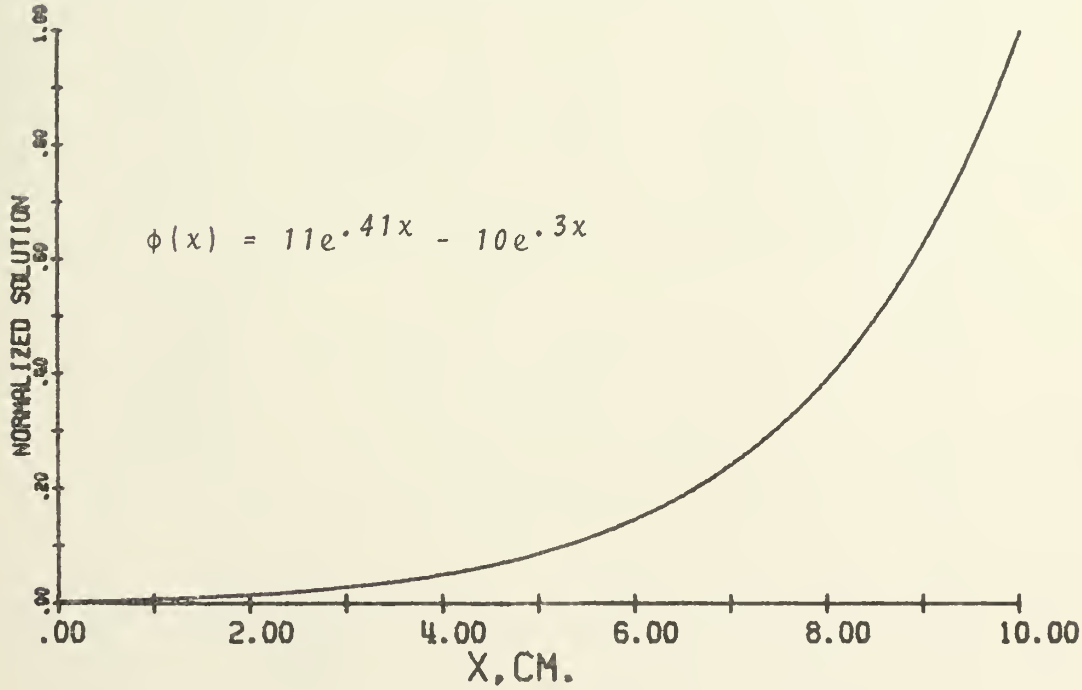


TABLE VIII

DISCRETIZATION ERROR IMPROVEMENT

Alpha	Number of Iterations to Converge	Fractional Error Near x = 0	Fractional Error Near x = 5	Fractional Error Near x = 10	Remarks
Unmodified S_N	11	.0011	.01173	.0306	Largest errors are at large-x.
0.5	11	.00056	.0000659	.00173	Vast error improvement for all x.
0.7	11	.000149	.000158	.0000259	Vast error improvement for all x.
0.9	11	.0000231	.0000449	.0000541	This value of α provided the best overall error improvement. Factors of 50/200/600 over unmodified S_N.
1.1	11	.00028	.00073	.00104	Errors here still better than for unmodified S_N.
1.3	11	.0129	.0039	.00616	Some small-x errors larger than unmodified S_N errors.

for the unmodified S_N algorithm and

$$\frac{|\phi_k^{exact} - \phi_k^{(i)}|}{\phi_k^{exact}} \tag{31}$$

for the transform domain algorithm. The errors compared, however, are for values of the unmodified S_N ψ-domain solution and the transform method ψ-domain solution. These fractional errors are attributed primarily to discretization error because the unit roundoff error for the double precision mode of the computer is 10^{-14} and the iterative error should be less than 10^{-3} due to the convergence criterion (25) used. Certainly fractional errors greater than 10^{-3}, as most unmodified S_N errors are, can be attributed to discretization error.

Discussion of Results

The transform method produced steady discretization error improvement with α for all space points up to $\alpha = 0.9$. This error improvement is due to the general decrease in curvature magnitude of the exact ϕ-domain solution compared to the exact ψ-domain solution over the slab width. The transform method can be used to diminish ρ in equation (11), thereby reducing the discretization error.

The value $\alpha = 1.1$ produced error improvement over the unmodified S_N solution but less improvement than did $\alpha = 0.9$. Values of $\alpha > 1.1$ produced larger discretization errors than did unmodified S_N in some regions of the slab. Observe that

the solution $\phi(x)$, Figure 4.4, has the least curvature magnitude at $\alpha = 0.9$. Also at $\alpha > \sigma^a/\mu_0 \simeq 0.9$, equation (28) reveals that $\phi(x)$ has a growing exponential in the second term which dominates the solution. This suggests that predicting an optimum value for α requires knowledge of the exact solution. Obviously this is not possible in more complicated practical problems but some guidance is available from the simplified problem.

Strategy for Determining Optimum α

A practical value for α may be obtained from the following argument. The method is only useful for predominantly absorbing media. These problems are characterized by solutions which "follow the source." That is, if the source distribution is a decaying exponential, the exact solution will have the form of a decaying exponential. Consequently a transform which changes the source to nearly a constant in the ϕ-domain would probably produce a more slowly varying ϕ-domain solution with accompanying smaller discretization error. For the problem treated here, that would be

$$q(x)e^{\alpha x} = e^{x(\alpha-\sigma^t)} = const,$$

when

$$\alpha = \sigma^t = 1.0 \ .$$

This is very close to the value σ^a/μ_0 which was found nearly optimum by knowledge of the exact solution. In the absence

of knowledge of $\phi(x)$, the above strategy based on source shape could be useful. Of course if partial insight into the expected solution is available from analytic solutions to similar problems or if the user's intuition is sharp, a value of α based on such considerations could prove fruitful.

Application to S_N

The S_N algorithm for the full-range slab problem, derived in Chapter II, is repeated here

$$\psi_{k+1,j}^{(i+1)} = \frac{e_j}{d_j}\psi_{k,j}^{(i+1)} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\psi_{k+1/2,n}^{(i)} + q_{k+1/2,j}\right\} \tag{32a}$$

for $\mu_j > 0$ and

$$\psi_{k,j}^{(i+1)} = \frac{e_j}{d_j}\psi_{k+1,j}^{(i+1)} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\psi_{k+1/2,n}^{(i)} + q_{k+1/2,j}\right\} \tag{32b}$$

for $\mu_j < 0$

where

$$e_j = \frac{2|\mu_j|-\Delta\sigma^t}{2\Delta} \tag{32c}$$

and

$$d_j = \frac{2|\mu_j|+\Delta\sigma^t}{2\Delta} \quad . \tag{32d}$$

The transform

$$\psi_j(x) = \phi_j(x)e^{-\alpha x} \tag{21}$$

with $\alpha > 0$ applied to equation (1) gives

$$\mu_j \frac{d\phi_j(x)}{dx} + (\sigma^t - \alpha\mu_j)\phi_j(x) = \frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\phi_n(x) + q_j(x)e^{\alpha x} \quad (33a)$$

for $\mu_j > 0$ and

$$-|\mu_j|\frac{d\phi_j(x)}{dx} + (\sigma^t + \alpha|\mu_j|)\phi_j(x) = \frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\phi_n(x) + q_j(x)e^{\alpha x} \quad (33b)$$

for $\mu_j < 0$. Observe that the transform adds absorption to the $\mu_j < 0$ equations but diminishes absorption in the $\mu_j > 0$ equations.

Spatial discretization is conducted in the manner previously described and coefficients of the fluxes are collected to give the S_N algorithm,

$$\phi_{k+1,j}^{(i+1)} = \frac{e_j}{d_j}\phi_{k,j}^{(i+1)} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\phi_{k+1/2,n}^{(i)} + q_{k+1/2,j}e^{\alpha x_{k+1/2}}\right\} \quad (34a)$$

for $\mu_j > 0$ and

$$\phi_{k,j}^{(i+1)} = \frac{e_j}{d_j}\phi_{k+1,j}^{(i+1)} + \frac{1}{d_j}\left\{\frac{\sigma^s}{2}\sum_{n=1}^{N}\omega_n\phi_{k+1/2,n}^{(i)} + q_{k+1/2,j}e^{\alpha x_{k+1/2}}\right\} \quad (34b)$$

for $\mu_j < 0$ where

$$e_j = \begin{cases} e_j + \dfrac{\alpha|\mu_j|}{2} & \text{for } \mu_j > 0 \quad (35a) \\[2ex] e_j - \dfrac{\alpha|\mu_j|}{2} & \text{for } \mu_j < 0 \quad (35b) \end{cases}$$

and

$$d_j = \begin{cases} d_j - \dfrac{\alpha|\mu_j|}{2} & \text{for } \mu_j > 0 \quad (36a) \\[2ex] d_j + \dfrac{\alpha|\mu_j|}{2} & \text{for } \mu_j < 0 \quad (36b) \end{cases}$$

Observe that α increases $\frac{e_j}{d_j}$ and $\frac{1}{d_j}$ for $\mu_j > 0$. Consequently the part of $(D-E)^{-1}S$ which corresponds to the $\mu_j > 0$ equations has α positioned such that each non-zero matrix element is increased. Conversely, α diminishes $\frac{e_j}{d_j}$ and $\frac{1}{d_j}$ for $\mu_j < 0$. This results in a decrease in the non-zero matrix elements of the iteration matrix corresponding to the $\mu_j < 0$ equations. These properties destroy the simple block collapsing technique, demonstrated in Chapter II, used to calculate the norm of the transform domain iteration matrix. Some portions of the maximum row sum are diminished and some are increased by α. Fortunately it will not be necessary to calculate $||(D-E)^{-1}S||$ for the problem treated here since only predominantly absorbing media are considered. Consequently the norm will not change significantly with α nor will the number of iterations to convergence.

Sample Problem

Equations (34), subject to the boundary conditions

$$\phi_{1,j} = 1.0 \quad \text{for } \mu_j > 0 \tag{37a}$$

and

$$\phi_{R+1,j} = 0 \quad \text{for } \mu_j < 0 \tag{37b}$$

and source

$$q_j(x) = 0 \quad \text{for all } j, \tag{38}$$

were solved on an IBM 360/67 computer. The pointwise convergence criterion

$$\frac{\left|\phi_k^{(i+1)} - \phi_k^{(i)}\right|}{\phi_k^{(i)}} < \varepsilon \tag{39}$$

was used. The problem parameters are:

Slab width = 10.0 cm.

R = 30 spatial intervals

$\sigma^t = 1.0\ \text{cm}^{-1}$.

$\sigma^s = 0.2\ \text{cm}^{-1}$.

Quadrature set = six point Gauss-Legendre

α = Various values from 0.1 to 1.0.

The resulting ψ-domain solution will be compared to two unmodified S_N solutions of the same problem, described by equations (32), but with different spatial mesh sizes. One used $R = 600$, the other $R = 30$. Appendix E contains the programs and sample output.

Unmodified S_N Results

Comparison of the unmodified S_N results for $R = 600$ and $R = 30$ shows that all $R = 600$ values are either equal to or greater than (at least to five significant figures) the $R = 30$ solution values. Since the $R = 600$ solution has less discretization error than the $R = 30$ solution, the above fact simplifies the analysis of the effect of the transform method on improving discretization error. If the transform method ψ-domain solution values are between the unmodified S_N $R = 30$ and $R = 600$ values, the method has improved the

discretization error. The above criterion is used to judge the effectiveness of the transform method in improving discretization error.

Transform Method Results

The data show that the transform method decreases the discretization error for the problem. Discretization error improvement is not pointwise uniform but, in general, transform method solution values approach the $R = 600$ solution values as α is increased. The bulk solution differences, displayed and defined in Table IX, diminish for each μ_j component of the solution as α increases. The total bulk difference also diminishes as α increases. This means that some integral measure of the discretization error diminishes as α increases through $\alpha = 1.0$. A closer look at the pointwise data, however, reveals that at $\alpha = 1.0$ many transform method ψ-domain solution values have overshot the unmodified S_N $R = 600$ solution values.

Table X displays a more detailed analysis of the pointwise discretization error improvement achieved by the transform method. It shows that discretization error is decreased at each point for values of α up to 0.5. For $\alpha > 0.5$, discretization error is generally improved but some values overshoot the $R = 600$ values. As α grows larger, more values overshoot the $R = 600$ values and the analysis loses its basis for comparison. All methods required the same number of iterations to converge; 10. Hence

TABLE IX

BULK SOLUTION DIFFERENCES FOR VARIOUS α

Alpha	MU(1)	MU(2)	MU(3)	MU(4)	MU(5)	MU(6)	Total
0*	0.10103	0.037411	0.026569	0.0040609	0.0024641	0.0019834	0.17352
0.1	0.09294	0.029666	0.019039	0.0035137	0.0021072	0.0016895	0.14896
0.3	0.078202	0.017448	0.0084172	0.0026179	0.0015301	0.0012161	0.10943
0.5	0.065267	0.0090181	0.0030947	0.0019507	0.0011095	0.00087339	0.081313
0.7	0.053990	0.0038904	0.0011895	0.0014677	0.0081394	0.00063493	0.061987
0.9	0.044222	0.0015471	0.00097341	0.0011245	0.00061195	0.00047407	0.048953
1.0	0.039854	0.0011347	0.00088316	0.0009915	0.00053619	0.00041443	0.043814

*Signifies unmodified S_N with R = 30 spatial intervals.

The bulk differences are defined by:

$$\text{Bulk difference for MU(J)} = \sum_{k} \left| (\text{Unmodified } R\text{=600 solution})_{k,j} - (\text{transform method } R\text{=30 solution})_{k,j} \right|$$

$$\text{Total bulk difference} = \sum_{j} \left(\text{bulk difference for MU(J)}\right)$$

TABLE X

DISCRETIZATION ERROR IMPROVEMENT

Alpha	Significant Figure of Closest Agreement with R = 600 Algorithm Solution	Analysis of Differences between Unmodified R = 600 Solution and Transform Method R = 30 Solution	General Remarks (Agreement Refers to Comparison with R = 600 Unmodified S_N Solution)
0*	2nd for $\mu > 0$ 5th for $\mu < 0$	All solution values are smaller than R = 600 values.	Agreement is generally only to first significant figure.
0.1	2nd for $\mu > 0$ 5th for $\mu < 0$	All values closer to R = 600 solution than are the unmodified R = 30 values.	Discretization error improved at each point. Agreement generally only to first significant figure.
0.3	3rd for $\mu > 0$ 5th for $\mu < 0$	All values closer to R = 600 solution than α = 0.1 values are.	Discretization error improved at each point. Agreement generally to 2nd significant figure for $\mu > 0$ values. Agreement generally to 1st significant figure for $\mu < 0$ values.
0.5	3rd for $\mu > 0$ 5th for $\mu < 0$	All values but 1 out of 186 have closer values to R = 600 solution than did α = 0.3 solution.	Discretization error improved at each point. One value overshot R = 600 value. Agreement generally to 2nd significant figure for $\mu > 0$ values. Agreement generally to 1st significant figure for $\mu < 0$ values.

TABLE X--Continued

Alpha	Significant Figure of Closest Agreement with R = 600 Algorithm Solution	Analysis of Differences between Unmodified R = 600 Solution and Transform Method R = 30 Solution	General Remarks (Agreement Refers to Comparison with R = 600 Unmodified S_N Solution)
0.7	3rd for $\mu > 0$ 5th for $\mu < 0$	All but 7 of 186 values have smaller differences from R = 600 values than did α = 0.5 solution.	Discretization error generally improved. A few values slightly overshot R = 600 solution. One value overshot by 8%. Many values agree to 3rd significant figure for $\mu > 0$. Agreement generally to 2nd significant figure for $\mu < 0$.
0.9	5th for $\mu > 0$ 5th for $\mu < 0$	Most differences from R = 600 result are smaller than are those for α = 0.7.	Majority of points have improved discretization error. A few values overshot R = 600 solution. One overshot 9.5%. Many values agree to 4th and most agree to 3rd significant figure for $\mu > 0$. Agreement generally to 3rd significant figure for $\mu < 0$.
1.0	4th for $\mu > 0$ 5th for $\mu < 0$	Overshoot on many values causes larger differences from R = 600 values than were exhibited for α = 0.9 results.	Many values (24 out of 186) overshot R = 600 values. One overshoot was 10.2%. Many values agree to 4th and most agree to 3rd significant figure for $\mu > 0$. Agreement generally to 3rd significant figure for $\mu < 0$.

discretization error improvement was achieved at the expense of only $R \cdot N$ extra computations.

Conclusion

The transform method can be used to decrease the discretization error in S_N problems. This method is effective when the transform domain solution has less curvature magnitude over the slab width than the unmodified S_N solution.

The transform used for this purpose must be tailored to each individual problem. The choice of a successful transform for a specific problem is based upon intuition developed for that problem by experience, knowledge of analytic solutions to similar problems, or guess. For problems in which solutions tend to "follow the source," a transform may be devised which converts the source to a near constant in the transform domain. A possible transform candidate is some variation of the integrating factor for the transport equation neglecting the scattering gain term.

CHAPTER V

ROUNDOFF ERROR ANALYSIS FOR ITERATIVE METHODS

Background

Previous chapters have discussed the concepts of iteration and discretization errors in the S_N method. Each effect was treated with regard to its role in the convergence of the algorithm to some exact solution. The approximate solution $\underline{\psi}^{(i)}$ was shown to approach the exact solution $\psi_j(x)$ of the angular segmented equations as i approaches ∞ and Δ approaches zero. This concept of convergence ignores the errors that occur in any machine computation due to the rounding procedure employed in floating point arithmetic. In order to be effective, however, an algorithm must remain immune to the accumulation of roundoff errors. This immunity is called numerical stability. An otherwise convergent algorithm may be driven divergent if it does not possess this property. In order to analyze the property of numerical stability, precise definitions must be made.

Well-Posed Computation

A numerical method, which produces a solution approximating the exact solution of the problem, is said to be a well-posed computation if the solution is insensitive

to small changes in data or to roundoff errors. Here data refers to the elements of matrices involved and the initial or boundary conditions and sources. Isaacson[53] defines a computing problem as an algorithm or "a set of rules specifying the order and kind of arithmetic operations (i.e. rounding rules) to be used on specified data." He also defines a computing problem to be well-posed if and only if

i) a solution exists

ii) the solution is unique, that is, when performed several times, with the same data, identical results are obtained

iii) the solution depends Lipschitz continuously on the data with a constant that is not too large. The last condition requires that "small" changes in the data must result in only "small" changes in the solution. A well-posed computation possesses the property of numerical stability. A necessary condition that a finite difference scheme be convergent is that it possess numerical stability.

Condition Number

Roundoff errors will in general be introduced in the course of carrying out the arithmetic required to solve the system of equations

$$A\underline{x} = \underline{b} \; . \tag{1}$$

But if the algorithm is well-posed, these errors can be kept within reasonable bounds. The matrix A is said to be

"well-conditioned" or "ill-conditioned" if the computation is or is not, respectively, well-posed. The condition number, $\nu(A)$, for A serves as a measure of ill-conditioning. The condition number is defined by

$$\nu(A) = ||A||\cdot||A^{-1}|| \ . \tag{2}$$

Some important properties of the condition number are:

i) $\nu(A) \geq 1$ with equality only if A is orthogonal.

ii) $\nu(A) = \nu(A^{-1})$.

iii) $\nu(\alpha A) = \nu(A)$.

Suppose that the data A and $\underline{b}$ of equation (1) have small perturbations or uncertainties δA and $\underline{\delta b}$ respectively due to roundoff error. The computer is solving the slightly perturbed system

$$(A + \delta A)(\underline{X} + \underline{\delta X}) = \underline{b} + \underline{\delta b} \tag{3}$$

instead of system (1). Isaacson[53] shows that if

$$||\delta A|| < 1/||A^{-1}|| \ , \tag{4}$$

the resulting relative uncertainty in the solution, $\underline{X}$, is

$$\frac{||\underline{\delta X}||}{||\underline{X}||} \leq \frac{\nu}{1-\nu\cdot\frac{||\delta A||}{||A||}}\left(\frac{||\underline{\delta b}||}{||\underline{b}||} + \frac{||\delta A||}{||A||}\right) . \tag{5}$$

Observe that if $\nu(A)$ is relatively small, A is well-conditioned and "small" relative uncertainties in the data produce "small" relative uncertainties in the solution. Conversely if $\nu(A)$ is very large, A is ill-conditioned and

"small" relative uncertainties in the data can destroy the solution.

Forsythe[64] points out a popular misconception that the smallness of $det(A)$ causes the ill-condition of A. That is, the condition number is not a measure of how nearly A is singular, hence difficult to invert. He stresses that the size of $\nu(A)$ is a far more important criterion of the "badness" of computational system (1) than either the smallness of $det(A)$ or the largeness of the order n of the system.

A theorem from Isaacson,[53] which will prove useful later, is stated here.

Theorem 2

If the matrix A has $||A|| < 1$, then $(I \pm A)$ is non-singular and

$$\frac{1}{1+||A||} \leq ||(I \pm A)^{-1}|| \leq \frac{1}{1-||A||} . \tag{6}$$

Application to Iterative Methods

Isaacson,[53] referring to iterative methods for solving systems of equations (1), states "One of the intrinsic advantages of such methods is the fact that errors, due to roundoff or even blunders, may be damped out as the iterative procedure continues". The following analysis provides an insight into the mechanism of this intrinsic advantage.

The interative S_N algorithm, described in Chapter II, has a matrix representation

$$(D-E)\underline{\psi}^{(i+1)} = S\underline{\psi}^{(i)} + \underline{q} \; . \tag{7}$$

At each iterative step, the data D, E, S, $\underline{q}$, and $\underline{\psi}^{(i)}$ are known and the machine is expected to invert (D-E) exactly and multiply the result onto the right-hand-side of (7), that is, solve for

$$\underline{\psi}^{(i+1)} = (D-E)^{-1}S\underline{\psi}^{(i)} + (D-E)^{-1}\underline{q} \tag{8}$$

precisely. The computer, however, has a limited storage capacity for each digit and must round off the input data to the last significant digit of its capacity. Consequently the uncertainty in the stationary data is δD, δE, δS, and $\underline{\delta q}$ which are of the order of the unit roundoff error of the machine (10^{-14} for the IBM 360/67 in double precision mode). These uncertainties are propagated through the arithmetic process of inverting (D-E) and matrix multiplying the result onto appropriate right hand side vectors. Consequently even the input data $\underline{\psi}^{(i)}$ has its uncertainty, $\underline{\delta\psi}^{(i)}$, carried with it in equation (7). Therefore the exact system to be solved at the $i+1^{st}$ iterate is not (7) but

$$(D-E)\underline{\psi}^{(i+1)} = S\underline{\psi}_c^{(i)} + \underline{q} \tag{9}$$

where the previously computed solution, $\underline{\psi}_c^{(i)}$, is

$$\underline{\psi}_c^{(i)} = \underline{\psi}^{(i)} + \underline{\delta\psi}^{(i)} \; . \tag{10}$$

But now the machine must solve the perturbed system

$$(D+\delta D-E-\delta E)\ (\underline{\psi}^{(i+1)}+\underline{\delta\psi}^{(i+1)}) = (S+\delta S)(\underline{\psi_c}^{(i)}+\underline{\delta\psi_c}^{(i)})+(\underline{q}+\delta q) \qquad (11)$$

where $\underline{\delta\psi}_c^{(i)}$ is the roundoff error resulting from machine handling of the data $\underline{\psi}_c^{(i)}$. It is the same order of magnitude as δD, δE, δS, and $\underline{\delta q}$. These uncertainties in the input data produce an uncertainty $\underline{\delta\psi}^{(i+1)}$ in the solution. Using equation (11), it is possible to derive an upper bound on the norm of the relative uncertainty in the solution

$$\frac{||\underline{\delta\psi}^{(i+1)}||}{||\underline{\psi}^{(i+1)}||} \text{ at each iterative step.}$$

First Iteration

Suppose the initial guess for algorithm (7) is

$$\underline{\psi}^{(0)} = 0 \qquad (12)$$

with no machine error. That is, the first machine iteration is on $\underline{q}$. The first iterate is

$$(D-E)\underline{\psi}^{(1)} = \underline{q} \qquad (13)$$

but the computer solves the perturbed system

$$(D+\delta D-E-\delta E)(\underline{\psi}^{(1)}+\underline{\delta\psi}^{(1)}) = \underline{q} + \underline{\delta q} \ . \qquad (14)$$

Factoring out (D-E) from the left-hand-side and multiplying by $(D-E)^{-1}$ gives

$$\left[\underline{I}+(D-E)^{-1}(\delta D-\delta E)\right]\left(\underline{\psi}^{(1)}+\underline{\delta\psi}^{(1)}\right) = (D-E)^{-1}(\underline{q}+\underline{\delta q}) \ . \qquad (15)$$

Recognizing that

$$\underline{\psi}^{(1)} = (D-E)^{-1}\underline{q} \tag{16}$$

from (13), cancelling this equality in (15) and collecting terms in $\underline{\delta\psi}^{(1)}$ on the left-hand-side produces

$$\left[I+(D-E)^{-1}(\delta D-\delta E)\right]\underline{\delta\psi}^{(1)} = (D-E)^{-1}(\delta E-\delta D)\underline{\psi}^{(1)}+(D-E)^{-1}\underline{\delta q}\,. \tag{17}$$

Hence

$$\underline{\delta\psi}^{(1)} = \left[I+(D-E)^{-1}(\delta D-\delta E)\right]^{-1}\left\{(D-E)(\delta E-\delta D)\underline{\psi}^{(1)}+(D-E)^{-1}\underline{\delta q}\right\}. \tag{18}$$

But due to Theorem 2,

$$\left|\left|\left[I+(D-E)^{-1}(\delta D-\delta E)\right]^{-1}\right|\right| \leq \frac{1}{1-||(D-E)^{-1}(\delta D-\delta E)||} \tag{19}$$

subject to the restriction that

$$||\delta D-\delta E|| < \frac{1}{||(D-E)^{-1}||}\,. \tag{20}$$

Hence

$$||\underline{\delta\psi}^{(1)}|| \leq \frac{1}{1-||(D-E)^{-1}(\delta D-\delta E)||}\left\{||(D-E)^{-1}||\cdot||\delta D-\delta E||\cdot||\underline{\psi}^{(1)}|| + ||(D-E)^{-1}||\cdot||\underline{\delta q}||\right\}.$$

Now divide by $||\underline{\psi}^{(1)}||$ and multiply by $\frac{||(D-E)||}{||(D-E)||}$ to obtain

$$\frac{||\underline{\delta\psi}^{(1)}||}{||\underline{\psi}^{(1)}||} \leq \frac{\nu}{1-||(D-E)^{-1}(\delta D-\delta E)||}\left\{\frac{||\delta D-\delta E||}{||(D-E)||} + \frac{||\underline{\delta q}||}{||(D-E)||\cdot||\underline{\psi}^{(1)}||}\right\}. \tag{21}$$

where ν is the condition number of the matrix (D-E), that is,

$$\nu = ||(D-E)||\cdot||(D-E)^{-1}|| \quad . \tag{22}$$

But from (13),

$$||\underline{q}|| \leq ||(D-E)||\cdot||\underline{\psi}^{(1)}||$$

or

$$\frac{1}{||D-E||\cdot||\underline{\psi}^{(1)}||} \leq \frac{1}{||\underline{q}||} \quad . \tag{23}$$

From restriction (20) and the fact that

$$||(D-E)^{-1}(\delta D-\delta E)|| \leq ||(D-E)^{-1}||\cdot||\delta D-\delta E|| < 1 \ ,$$

$$1-||(D-E)^{-1}(\delta D-\delta E)|| \geq 1-||(D-E)^{-1}||\cdot||\delta D-\delta E|| \quad \text{hence}$$

$$\frac{1}{1-||(D-E)^{-1}(\delta D-\delta E)||} \leq \frac{1}{1-||(D-E)^{-1}||\cdot||\delta D-\delta E||} \tag{24}$$

Using (23) and (24), inequality (21) becomes

$$\frac{||\delta\underline{\psi}^{(1)}||}{||\underline{\psi}^{(1)}||} \leq \frac{\nu}{1-\nu\dfrac{||\delta D-\delta E||}{||D-E||}}\left\{\frac{||\delta D-\delta E||}{||D-E||} + \frac{||\delta\underline{q}||}{||\underline{q}||}\right\} \quad . \tag{25}$$

Second Iteration

The second iteration treats the problem

$$(D-E)\underline{\psi}^{(2)} = S\underline{\psi}_c^{(1)} + \underline{q} \tag{26}$$

but the machine solves the perturbed system

$$(D+\delta D-E-\delta E)\left(\underline{\psi}^{(2)}+\underline{\delta\psi}^{(2)}\right) = (S+\delta S)\left(\underline{\psi}_c^{(1)}+\delta\underline{\psi}_c^{(1)}\right)+(\underline{q}+\underline{\delta q}) \ . \tag{27}$$

Following the same procedure used during the first iteration,

$$\frac{||\delta\underline{\psi}^{(2)}||}{||\underline{\psi}^{(2)}||} \leq \frac{\nu}{1-\nu\cdot\frac{||\delta D-\delta E||}{||D-E||}}\left\{\frac{||\delta E-\delta D||}{||D-E||} + \frac{||\delta S||}{||D-E||}\cdot\frac{||\underline{\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||} + \frac{||S+\delta S||}{||D-E||}\cdot\frac{||\delta\underline{\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||} + \frac{||\delta\underline{q}||}{||D-E||\cdot||\underline{\psi}^{(2)}||}\right\} . \qquad (28)$$

A bound on the quantity $\frac{||\underline{\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||}$ can be derived from (26). That is,

$$\underline{\psi}^{(2)} = (D-E)^{-1}S\underline{\psi}_c^{(1)} + (D-E)^{-1}\underline{q}$$

or

$$\underline{\psi}_c^{(1)} = \left[(D-E)^{-1}S\right]^{-1}\left[\underline{\psi}^{(2)} - (D-E)^{-1}\underline{q}\right] .$$

Consequently

$$||\underline{\psi}_c^{(1)}|| \leq \left|\left|\left[(D-E)^{-1}S\right]^{-1}\right|\right| \left[||\underline{\psi}^{(2)}|| + ||(D-E)^{-1}||\cdot||\underline{q}||\right]$$

or

$$\frac{||\underline{\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||} \leq \left|\left|\left[(D-E)^{-1}S\right]^{-1}\right|\right| \left[1 + \frac{||(D-E)^{-1}||\cdot||\underline{q}||}{||\underline{\psi}^{(2)}||}\right] . \qquad (29)$$

A bound on $\frac{1}{||\underline{\psi}^{(2)}||}$ on the R.H.S. of (29) can be derived by using the unperturbed system

$$\underline{\psi}^{(2)} = (D-E)^{-1}S\underline{\psi}^{(1)} + (D-E)^{-1}\underline{q} . \qquad (30)$$

From equation (13)

$$\underline{\psi}^{(1)} = (D-E)^{-1}\underline{q} \qquad (31)$$

which substituted into (30) gives

$$\left[I + (D-E)^{-1}S\right]\underline{\psi}^{(1)} = \underline{\psi}^{(2)}$$

or

$$\underline{\psi}^{(1)} = \left[I + (D-E)^{-1}S\right]^{-1}\underline{\psi}^{(2)} \quad . \tag{32}$$

Using Theorem 2,

$$||\underline{\psi}^{(1)}|| \leq \frac{||\underline{\psi}^{(2)}||}{1-||(D-E)^{-1}S||}$$

subject to the restriction that

$$||(D-E)^{-1}S|| < 1 \quad . \tag{33}$$

But recall that (33) is the condition that the S_N algorithm must meet to guarantee convergence so it is met a priori. Consequently

$$\frac{1}{||\underline{\psi}^{(2)}||} \leq \frac{1}{(1-||(D-E)^{-1}S||)\cdot||\underline{\psi}^{(1)}||} \quad .$$ But from (23),

$$\frac{1}{||\underline{\psi}^{(1)}||} \leq \frac{||D-E||}{||\underline{q}||}$$ so the above inequality becomes

$$\frac{1}{||\underline{\psi}^{(2)}||} \leq \frac{||D-E||}{||\underline{q}||\cdot(1-||(D-E)^{-1}S||)} \quad . \tag{34}$$

Using (34) on the R.H.S. of (29) produces

$$\frac{||\underline{\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||} \leq \left|\left|\left[(D-E)^{-1}S\right]^{-1}\right|\right| \left[\frac{1+\nu-||(D-E)^{-1}S||}{1-||(D-E)^{-1}S||}\right] \quad . \tag{35}$$

Also using (34) for the factor $\dfrac{||\underline{\delta\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||}$ in (28) provides the bound

$$\frac{||\underline{\delta\psi}_c^{(1)}||}{||\underline{\psi}^{(2)}||} \le \frac{||D-E||\cdot||\underline{\delta\psi}_c^{(1)}||}{||\underline{q}||\cdot(1-||(D-E)^{-1}S||)} \quad . \tag{36}$$

Now applying (34), (35), and (36) to (28) produces

$$\frac{||\underline{\delta\psi}^{(2)}||}{||\underline{\psi}^{(2)}||} \le \frac{\nu}{1-\nu\cdot\frac{||\delta D-\delta E||}{||D-E||}}\left\{\frac{||\delta E-\delta D||}{||D-E||} + \frac{1}{\left[1-||(D-E)^{-1}S||\right]}\left[\frac{||\delta S||}{||D-E||}\cdot\frac{\rho(1+\nu-||(D-E)^{-1}S||)}{||(D-E)^{-1}S||} + \right.\right.$$

$$\left.\left. + \frac{||S+\delta S||}{||\underline{q}||}\cdot||\underline{\delta\psi}_c^{(1)}|| + \frac{||\underline{\delta q}||}{||\underline{q}||}\right]\right\} \tag{37}$$

where

$$\rho = ||(D-E)^{-1}S||\cdot||\left[(D-E)^{-1}S\right]^{-1}|| \tag{38}$$

is the condition number of the iteration matrix.

The General Iterate

Suppose the i^{th} iteration has been completed. That is, $\underline{\psi}_c^{(i)} = \underline{\psi}^{(i)} + \underline{\delta\psi}^{(i)}$ has been computed. A bound on the relative uncertainty in the $i+1^{st}$ iterate solution can be derived from recognizing that instead of solving the system

$$(D-E)\underline{\psi}^{(i+1)} = S\underline{\psi}_c^{(i)} + \underline{q} \tag{39}$$

the machine solves the perturbed system

$$(D+\delta D-E-\delta E)\left(\underline{\psi}^{(i+1)}+\underline{\delta\psi}^{(i+1)}\right) = (S+\delta S)\left(\underline{\psi}_c^{(i)}+\underline{\delta\psi}_c^{(i)}\right) + (\underline{q}+\underline{\delta q}) \quad . \tag{40}$$

Proceeding in the same fashion as in the previous section,

$$\frac{\|\underline{\delta\psi}^{(i+1)}\|}{\|\underline{\psi}^{(i+1)}\|} \le \frac{\nu}{1-\nu\cdot\frac{\|\delta D-\delta E\|}{\|D-E\|}} \left\{ \frac{\|\delta E-\delta D\|}{\|D-E\|} + \frac{\|\delta S\|}{\|D-E\|} \cdot \frac{\|\underline{\psi}_c^{(i)}\|}{\|\underline{\psi}^{(i+1)}\|} + \frac{\|S+\delta S\|}{\|D-E\|} \cdot \frac{\|\underline{\delta\psi}_c^{(i)}\|}{\|\underline{\psi}^{(i+1)}\|} + \right.$$
$$\left. + \frac{\|\underline{\delta q}\|}{\|D-E\|\cdot\|\underline{\psi}^{(i+1)}\|} \right\} . \qquad (41)$$

From equation (39),

$$\underline{\psi}^{(i+1)} = (D-E)^{-1} S\underline{\psi}_c^{(i)} + (D-E)^{-1}\underline{q}$$

or

$$\underline{\psi}_c^{(i)} = \left[(D-E)^{-1}S\right]^{-1}\left[\underline{\psi}^{(i+1)} - (D-E)^{-1}\underline{q}\right] .$$

Consequently

$$\frac{\|\underline{\psi}_c^{(i)}\|}{\|\underline{\psi}^{(i+1)}\|} \le \left\|\left[(D-E)^{-1}S\right]^{-1}\right\| \cdot \left[1 + \frac{\|(D-E)^{-1}\|\cdot\|\underline{q}\|}{\|\underline{\psi}^{(i+1)}\|}\right] \qquad (42)$$

The factor $\frac{1}{\|\underline{\psi}^{(i+1)}\|}$ on the R.H.S. of (42) can be bounded by recalling that the unperturbed system obeys

$$\underline{\psi}^{(i+1)} = \sum_{m=0}^{i} A^m (D-E)^{-1}\underline{q} \qquad (43)$$

where for convenience of notation

$$A = (D-E)^{-1}S . \qquad (44)$$

Consequently

$$\underline{q} = (D-E)\left[I + A + A^2 + \cdots + A^i\right]^{-1}\underline{\psi}^{(i+1)} .$$

But if condition (33) is met,

$$(I-A)^{-1} = \sum_{m=0}^{\infty} A^m$$

so

$$\begin{aligned}\left[I+A+A^2+\cdots+A^i\right] &= \left[(I-A)^{-1} - A^{i+1} - A^{i+2} \cdots\right]^{-1} \\ &= \left[(I-A)^{-1} - A^{i+1}\left(I+A+A^2+\cdots\right)\right]^{-1} \\ &= \left[\left(I-A^{i+1}\right)(I-A)^{-1}\right]^{-1} \\ &= (I-A)\left(I-A^{i+1}\right)^{-1} .\end{aligned}$$

Hence

$$q = (D-E)(I-A)\left(I-A^{i+1}\right)^{-1}\underline{\psi}^{(i+1)} .$$

Consequently

$$||\underline{q}|| \leq \frac{||D-E||\cdot\left(1+||A||\right)\cdot||\underline{\psi}^{(i+1)}||}{1 - ||A^{i+1}||} \tag{45}$$

due to Theorem 2 if

$$||A^{i+1}|| < 1$$

which is certainly true if condition (33) is met. Since

$$||A^{i+1}|| \leq ||A||^{i+1} < 1 ,$$

$$1-||A^{i+1}|| \geq 1-||A||^{i+1}$$

and

$$\frac{1}{1-||A^{i+1}||} \leq \frac{1}{1-||A||^{i+1}} \quad . \tag{46}$$

Using (46), inequality (45) becomes

$$||\underline{q}|| \leq \frac{||D-E||\cdot(1+||A||)\cdot||\underline{\psi}^{(i+1)}||}{1-||A||^{i+1}}$$

or in terms of the iteration matrix norm

$$\frac{1}{||\underline{\psi}^{(i+1)}||} \leq \frac{||D-E||\cdot[1+||(D-E)^{-1}S||]}{||\underline{q}||\cdot\left[1-||(D-E)^{-1}S||^{i+1}\right]} \quad . \tag{47}$$

Applying (47) to the R.H.S. of (42) gives

$$\frac{||\underline{\psi}_c^{(i)}||}{||\underline{\psi}^{(i+1)}||} \leq \frac{\rho}{||(D-E)^{-1}S||}\left[\frac{1+\nu\left(1+||(D-E)^{-1}S||\right)-||(D-E)^{-1}S||^{i+1}}{1-||(D-E)^{-1}S||^{i+1}}\right] . \tag{48}$$

Now applying (47) and (48) to (41) produces

$$\frac{||\delta\underline{\psi}^{(i+1)}||}{||\underline{\psi}^{(i+1)}||} \leq \frac{\nu}{1-\frac{\nu\cdot||\delta D-\delta E||}{||D-E||}}\left\{\frac{||\delta E-\delta D||}{||D-E||}+\frac{1}{1-||(D-E)^{-1}S||^{i+1}}\left[\frac{||\delta S||}{||D-E||}\cdot\frac{\rho\left[1+\nu\left(1+||(D-E)^{-1}S||\right)-||(D-E)^{-1}S||^{i+1}\right]}{||(D-E)^{-1}S||}+\right.\right.$$
$$\left.\left.+\frac{||S+\delta S||}{||\underline{q}||}\cdot\left[1+||(D-E)^{-1}S||\right]\cdot||\delta\underline{\psi}_c^{(i)}||+\left[1+||(D-E)^{-1}S||\right]\cdot\frac{||\delta\underline{q}||}{||\underline{q}||}\right]\right\} . \tag{49}$$

This expression is valid for $i = 2,3,\cdots$.

Analysis of Error Bound

Observe that in expression (49), the only terms which are affected by the number of iterations are $||(D-E)^{-1}S||^{i+1}$ and $\frac{1}{1-||(D-E)^{-1}S||^{i+1}}$. As i increases, the former magnifies

the bound slightly and the latter diminishes the bound. The latter effect is predominant. Therefore the bound becomes tighter as iteration proceeds and roundoff error propagation is diminished. This effect is not strong enough to force the relative uncertainty to vanish, however, as i approaches ∞.

Certain terms and factors in (49) can be neglected with respect to more dominant quantities in order to obtain a qualitative view of the bound. That is,

$$\frac{||\delta\underline{\psi}^{(i+1)}||}{||\underline{\psi}^{(i+1)}||} \le \frac{\nu}{1-\nu\cdot\frac{||\delta D-\delta E||}{||D-E||}}\left\{\frac{||\delta E-\delta D||}{||D-E||}+\frac{||\delta S||}{||D-E||}\cdot\frac{\rho\nu}{||(D-E)^{-1}S||}+\frac{||S+\delta S||}{||\underline{q}||}\cdot||\delta\underline{\psi}_c^{(i)}||+\frac{||\delta\underline{q}||}{||\underline{q}||}\right\} \quad (50)$$

contains the dominant quantities in (49).

The condition of the matrix (D-E) has a dominant role in the roundoff error bound. If ν is very large, the relative uncertainty in the iterate solution will be significant. If ν is large enough that the factor $\nu\cdot\frac{||\delta D-\delta E||}{||D-E||}$ is close to unity, the relative uncertainty can be extremely large. For this to occur, ν would have to be of the order of 10^{14} for the double precision mode, extremely ill-conditioned. This event is unlikely. Since ν, which is always greater than unity, appears to the second power in (50), it is the dominant quantity in the bound.

The condition number of the iteration matrix, ρ, also plays an important part in the roundoff error bound. Its effect is subordinate to that of ν since ρ appears only to the first power. But if $(D-E)^{-1}S$ and (D-E) are both

ill-conditioned, the quantity $\nu^2\rho$ could cause significant roundoff error propagation in the solution.

The error bounds derived here display another important effect. Expressions (25), (37), and (49) each contain the uncertainty $||\delta D-\delta E||$ in important positions. This can cause the exact cancellation of roundoff errors during the iterative process. That is, $||\delta D-\delta E||$ vanishes. This gives vivid corroboration to Isaacson's observation quoted previously. The presence of $||\delta D-\delta E||$ in the dominating factor $\dfrac{\nu}{1-\dfrac{\nu\cdot||\delta D-\delta E||}{||D-E||}}$ reduces it to ν when exact cancellation occurs. This effect is also felt through the term $\dfrac{||\delta E-\delta D||}{||D-E||}$.

Observe that each term in the braces of (49) has a multiplier which is extremely small. That is δE, δD, δS, $\delta \underline{q}$, and $\delta \underline{\psi}_c^{(i)}$ are all of the order of the machine unit roundoff error. The factor $||\underline{q}||$ is generally on the order of unity. From equations (23), (24), (25), and (26) of Chapter II, the infinity norm of S is

$$||S|| = \frac{2\sigma^s}{4}\sum_{n=1}^{N}\omega_n$$

$$= \sigma^s \, . \tag{51}$$

From equations (16c), (16d), (21), and (22) of Chapter II, the infinity-norm of (D-E) is

$$||D-E|| = \max_j \left\{ |e_j| + d_j \right\}$$

$$= \max_j \frac{2|\mu_j|}{\Delta}$$

$$= \frac{2|\mu_j|_{max}}{\Delta} . \qquad (52)$$

As a consequence of the above facts, the quantity

$$\frac{\nu^2 \rho}{1 - \frac{\nu \cdot ||\delta D - \delta E||}{||D-E||}}$$

must be of the order of 10^{14} for the relative uncertainty to be significant. Assuming that $||\delta D - \delta E|| = 0$ and $\rho \simeq \nu$, this means that

$$\nu^3 \simeq 10^{14}$$

or

$$\nu \simeq 10^5$$

before significant roundoff error is propagated to the iterate solution. These extremely large condition numbers simply do not exist for physical problems of interest in the application of the S_N method.

Estimating the Condition Numbers

The condition number of the iteration matrix is not readily estimated. Although the infinity-norm of $(D-E)^{-1}S$ can be calculated precisely, as discussed in Chapter II, calculation of $||[(D-E)^{-1}S]^{-1}||$ is not tractable.

Consequently an estimate of ρ is not available nor is an upper bound on it. The following discussion assumes that ρ is small enough that its influence on the roundoff error bound is less than that of ν.

Estimation of ν is possible. Equations (27) and (28) in Chapter II reveal that the infinity norm of $(D-E)^{-1}$ is

$$||(D-E)^{-1}|| = \max_{j}\left\{\frac{1}{d_j}\left[1 + \left(\frac{|e_j|}{d_j}\right) + \left(\frac{|e_j|}{d_j}\right)^2 + \cdots + \left(\frac{|e_j|}{d_j}\right)^{R-1}\right]\right\} \tag{53a}$$

or

$$||(D-E)^{-1}|| = \frac{1}{d_i}\left[\frac{1-\beta^R}{1-\beta}\right] \tag{53b}$$

where R is the number of equi-spaced spatial intervals, Δ, and

$$0 \leq \beta = \frac{|e_i|}{d_i} < 1 . \tag{54}$$

Here

$$e_i = \frac{2|\mu_j|_{min} - \Delta\sigma^t}{2\Delta} \tag{55a}$$

and

$$d_i = \frac{2|\mu_j|_{min} + \Delta\sigma^t}{2\Delta} . \tag{55b}$$

From (53),

$$||(D-E)^{-1}|| < \frac{1}{d_i(1-\beta)} = \frac{1}{\sigma^t} . \tag{56}$$

Consequently from (52) and (53)

$$\nu = \frac{2|\mu_j|_{max}}{\Delta d_i}\left[\frac{1-\beta^R}{1-\beta}\right] \tag{57}$$

where i is the index for the minimum absolute ordinate in the set $\{\mu_j\}$.

A tight bound on ν can be obtained by neglecting β^R. That is,

$$\nu < \frac{2|\mu_j|_{max}}{\Delta\sigma^t} \quad .$$

Since all problems must have $|\mu_j|_{max} < 1$,

$$\nu < \frac{2}{\Delta\sigma^t} \tag{58}$$

is a useful estimator for ν. For the problems considered in Chapter II, $\sigma^t = 1.0\ cm^{-1}$ and $\Delta = 1/3\ cm$ so

$$\nu < 6 \quad . \tag{59}$$

Hence (D-E) is a very well-conditioned matrix for the problem considered. Consequently roundoff error propagation was insignificant in the investigations conducted. In practice, roundoff error propagation has not been experienced by this investigator which renders the S_N algorithm unstable. This must be a consequence of the well-conditioned state of the matrices $(D\text{-}E)^{-1}S$ and (D-E) as well as to the roundoff error cancellation property of the iterative method.

Effect of Exponential Transform

The presentation in Chapter III reveals that the transform

$$\psi_j(x) = \phi_j(x)e^{\alpha x} \tag{60}$$

applied to the S_N equations, produces an iterative algorithm with the same form as that of the unmodified S_N algorithm but with a modified source and a different iteration matrix, $(\mathcal{D}-\mathcal{E})^{-1}S$. As a consequence, the roundoff error analysis, previously presented, carries over into the transform domain in its entirety. That is, expressions (25), (37), and (49) are valid for

$$\frac{||\underline{\delta\phi}^{(i+1)}||}{||\underline{\phi}^{(i+1)}||}$$

if (D-E) and $(D-E)^{-1}S$ are replaced by $(\mathcal{D}-\mathcal{E})$ and $(\mathcal{D}-\mathcal{E})^{-1}S$ respectively and $\underline{q}$ is replaced by its transformed counterpart. All of the above quantities are defined in Chapter III.

From equations (18) and (19) of Chapter III it is clear that

$$||(\mathcal{D}-\mathcal{E})|| = ||(D-E)|| \tag{61}$$

since terms involving α cancel. From equations (23) and (24) of Chapter III,

$$||(\mathcal{D}-\mathcal{E})^{-1}|| = \frac{1}{d_i}\left[\frac{1-\gamma^R}{1-\gamma}\right] \tag{62}$$

where

$$0 \leq \gamma = \frac{|e_i|}{d_i} < 1 \tag{63}$$

and

$$e_i = \frac{2|\mu_j|_{min} - \Delta\sigma^t - \Delta\alpha|\mu_j|_{min}}{2\Delta} \tag{64a}$$

$$d_i = \frac{2|\mu_j|_{min} + \Delta\sigma^t + \Delta\alpha|\mu_j|_{min}}{2\Delta} . \tag{64b}$$

Observe that in problems for which the transform method is effective, i.e. $e_j \geq 0$ for all j, and

$$\frac{1}{d_j} < \frac{1}{\mathrm{d}_j} \tag{65}$$

of equation (55b) and

$$\gamma < \beta \tag{66}$$

of equation (54). Since $\frac{1}{d_i}$ is the predominant factor in (62),

$$||(D-E)^{-1}|| < ||(\mathrm{D}-\mathrm{E})^{-1}|| . \tag{67}$$

Consequently the condition number of $(D-E)$ is smaller than the condition number of (D-E). As previously discussed, the condition number of this matrix is the predominant factor in the stability of the iterative algorithm. Consequently it is expected that the transform domain iterative algorithm is "better" conditioned than the unmodified S_N algorithm.

CHAPTER VI

CONCLUSIONS

The S_N algorithm in one-dimensional plane geometry has been formulated within a mathematical framework which allows detailed insight into its convergence properties. Knowledge of these properties permits the derivation of an improved convergence criterion which, if met, guarantees that the fractional iterative error is arbitrarily small. Utilization of this criterion,

$$\frac{||\underline{\delta}^{(i+1)}||}{\psi_k^{(i)}} < \frac{\varepsilon}{1-\varepsilon}\left[1-||(D-E)^{-1}S||\right] \tag{1}$$

is especially important in problems for predominantly scattering media for which $||(D-E)^{-1}S||$ is close to unity.

The matrix formulation of the iterative S_N algorithm allows calculation of the infinity-norm of the iteration matrix, $||(D-E)^{-1}S||$, for homogeneous media problems in which scattering is isotropic in the Laboratory Coordinate System. Imposing the condition

$$||(D-E)^{-1}S|| < 1 \tag{2}$$

leads to sufficient conditions for which convergence of the S_N algorithm is guaranteed. These conditions impose

constraints on the values used for the numerical quantities of the problem, Δ, $|\mu_j|_{min}$, σ^s, and σ^t. The constraint

$$\Delta < \frac{|\mu_j|_{min}}{\sigma^s} \tag{3}$$

is sufficient to guarantee convergence and allows a limited amount of negativity in the matrix elements of the iteration matrix. The constraint

$$\Delta < \frac{|\mu_j|_{min}}{\sigma^t} \tag{4}$$

renders the iteration matrix non-negative and is sufficient to guarantee convergence for all physically realizable problems. The solution for such a system is non-negative for non-negative sources.

A spatial transform of the type

$$\psi_j(x) = \phi_j(x) f^{\alpha}(x) \tag{5}$$

was used on a special half-range problem. This transform places the acceleration parameter, α, in positions within the iteration matrix such that each non-zero matrix element is reduced. This diminishes the norm of the iteration matrix and accelerates convergence of the S_N algorithm. Reductions of almost 25% in the number of iterations to convergence have been realized for sample problems using the exponential transform

$$\psi_j(x) = \phi_j(x) e^{\alpha x} \tag{6}$$

where $\alpha > 0$. A practical maximum value of α for this transform was found to be

$$\alpha^* = \frac{2(\mu_j)_{min} - \Delta\sigma^t}{\Delta(\mu_j)_{min}} \; . \tag{7}$$

A practical useful value of α is much smaller than α^* and is chosen as a compromise between the user's estimate of tolerable discretization error and the reduction in calculational effort. It is possible that S_N problems in higher dimensional geometries can be formulated in the mathematical context displayed in this investigation for plane geometry. If transforms can be found which leave the angular discretized S_N equations invariant and reduce the norm of the iteration matrix, vast savings of computational effort can be realized for these problems. Possible transform candidates are some form of the integrating factor for the angular discretized equations describing the problem after neglecting the scattering gain term.

Nothing done here suggests that a transform can be found which is useful for all problems in a particular geometry. Experience of this investigator indicates that it is more likely that each problem in the geometry has a "best" transform which can be prescribed based upon the source, boundary conditions, and the prescriber's estimate of the form of the solution.

The spatial discretization error in the S_N algorithm is proportional to the second derivative of the solution $\psi_j(x)$. The spatial transform

$$\psi_j(x) = \phi_j(x)e^{-\alpha x} \tag{8}$$

with $\alpha > 0$ was used on a one-dimensional problem to decrease the discretization error. Discretization errors are shown to decrease because the ϕ-domain solution has less curvature magnitude than the unmodified S_N ψ-domain solution.

The transform used to decrease discretization errors must be tailored to each individual problem. The choice of a successful transform for a specific problem must be based on intuition developed for that problem by experience, knowledge of analytic solutions to similar problems, or guess. A possible candidate is some variation of the integrating factor for the problem.

An upper bound was derived for the relative uncertainty in the iterate solution of the S_N algorithm due to roundoff errors. This bound is expressed in terms of the condition numbers of the matrices (D-E) and $(D-E)^{-1}S$. The bound reveals that roundoff error cancellation can occur in the iterative process due to the factor $||\delta D-\delta E||$. If cancellation does not occur, however, roundoff errors can be magnified during the iterative process if the matrices $(D-E)^{-1}S$ and (D-E) are extremely ill-conditioned. Such an event is improbable. The bound on the fractional uncertainty for the iterate solution diminishes as iteration proceeds. It does not, however, vanish in the limit of an infinite number of iterations.

For the problems treated here, (D-E) is well-conditioned and its condition number is calculable from the problem parameters. The condition number of the iteration matrix is not calculable because $\left\| \left[(D-E)^{-1} S \right]^{-1} \right\|$ is not calculable. In the problems treated here, propagation of roundoff error was not significant.

REFERENCES

1. Davison, B., Neutron Transport Theory, Oxford University Press, London (1957).

2. Meghreblian, Robert V., and David K. Holmes, Reactor Analysis, McGraw-Hill, New York (1960).

3. Weinberg, Alvin M., and Eugene P. Wigner, The Physical Theory of Neutron Chain Reactors, University of Chicago Press, Chicago, Illinois (1958).

4. Case, Kenneth M., F. de Hoffman, and G. Placzek, Introduction to the Theory of Neutron Diffusion, Vol. I, U.S. Government Printing Office, Washington, D.C. (1954).

5. Case, Kenneth M., and Paul F. Zweiffel, Linear Transport Theory, Addison-Wesley, Reading, Massachusetts (1967).

6. Gelbard, Ely M., Computing Methods in Reactor Physics, edited by Greenspan, H., et al., Chapter 4, Gordon and Breach, New York (1968).

7. Ibid., Kalos, M. H., et al., Chapter 5.

8. Ibid., Carlson, Bengt, G., and K. D. Lathrop, Chapter 3.

9. Goldstein, Herbert and J. Ernest Wilkins, Jr., Calculations of the Penetration of Gamma Rays, NYO-3075, U.S. Atomic Energy Commission (1954).

10. Lathrop, Kaye D., "Ray Effects in Discrete Ordinate Equations," Nuclear Science and Engineering, 32, 357 (1968).

11. Wick, G. C., "Über ebene Diffusion Probleme," Z. Phys., 121, 707 (1943).

12. Chandrasekhar, S., Astrophysics Journal, 100, 76 (1944).

13. Chandrasekhar, S., Radiative Transfer, Oxford (1950).

14. Carlson, Bengt G., "Solution of Transport Equation by S_N Approximations," Los Alamos Scientific Laboratory Report LA-1599 (1953).

15. Carlson, Bengt G., "Development of the S_N Discrete Ordinate Method," A Review of the Discrete Ordinates S_N Method for Radiation Transport Calculations, edited by Trubey, D. K. and B. Masceiwicz, ORNL-RSIC-19 (March 1968).

16. Hendry, W. L., K. D. Lathrop, Sue Vandervoort, John Wooten, Bibliography of Neutral Particle Transport Theory, LA-4287-MS (September 1970).

17. Chernick, Jack, "Status of Reactor-Physics Calculations for U.S. Power Reactors," Reactor Technology, 13, 4, 368 (winter 1970-71).

18. Nakamura, Shoichiro, "MS_N, A New Approach to Neutron Transport Analysis," Nuclear Science and Engineering, 34, 83 (1968).

19. Shreiner, S., and D. S. Selengut, "Variational Development of S_N Theory," International Conference Research Reactors Utilization and Reactor Mathematics, Mexico (1967).

20. Kaplan, Stanley, J. A. Davis, and M. Natelson, "Angle-Space Synthesis - An Approach to Transport Approximations," Nuclear Science and Engineering, 28, 364 (1967).

21. Kaplan, Stanley, "A New Derivation of Discrete Ordinate Approximations," Nuclear Science and Engineering, 34, 76 (1968).

22. Natelson, M., "Variational Derivation of Discrete Ordinate-Like Approximations," Nuclear Science and Engineering, 43, 2, 131 (February 1971).

23. Lee, C. E., "The Discrete S_N Approximation to Transport Theory," Los Alamos Scientific Laboratory Report LA-2595, (1962).

24. Mills, Carroll B., "Fast Spectrum Breeder Reactor Analysis," Nuclear Applications, 5, 211 (1968).

25. Protsik, R., "Transport vs. Diffusion Theory in Large Fast Reactor Calculations," A Review of the Discrete Ordinates S_N Method for Radiation Transport Calculations, edited by Trubey, D. K. and B. Masceiwicz, ORNL-RSIC-19 (March 1968).

26. Carvik, I., "Monoenergetic Critical Parameters and Decay Constants for Small Homogeneous Spheres and Thin Homogeneous Slabs," Nuclear Science and Engineering, 31, 295 (1968).

27. Doyas, R. J. and B. L. Koponen, "Linearly Anisotropic Extensions of Asymptotic Neutron Diffusion Theory," Nuclear Science and Engineering, 41, 226 (1970).

28. Schwetje, W., "Synthesis Solutions of the Multigroup Boltzmann Equation for Slabs," Nuclear Science and Engineering, 46, 1, 159 (October 1971).

29. Burgart, Calvin E. and P. N. Stevens, "A General Method of Importance Sampling the Angle of Scattering in Monte Carlo Calculations," Nuclear Science and Engineering, 42, 306 (1970).

30. Reed, Wm. H., "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," Second Conference on Transport Theory, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (April 1971).

31. Reed, Wm. H. and K. D. Lathrop, "Truncation Error Analysis of Finite Difference Approximations to the Transport Equation," Nuclear Science and Engineering, 41, 237 (1970).

32. Richtmeyer, R. D. and M. Morton, Difference Methods for Initial Value Problems, Chapter 9, 2nd Edition, Interscience (1967).

33. Keller, H. B., "Approximate Solutions of Transport Problems. Part II, Convergence and Applications of the Discrete Ordinate Method," SIAM, 8, 43 (1960).

34. Wendroff, B., "On the Convergence of the Discrete Ordinate Method," SIAM, 8, 508 (1960).

35. Keller, H. B., "On the Pointwise Convergence of the Discrete Ordinate Method," SIAM, 8, 560 (1960).

36. Madson, Niel K., "Convergence of Difference Methods for the Linear Transport Equation," Technical Note BN-570, Institute for Fluid Dynamics and Applied Mathematics, University of Maryland, College Park, Md. (August 1968).

37. Madson, Niel K., "Discretization Error Estimates for Difference Approximations to Solutions of Transport Problems," WAPD-T-2236 (June 1969).

38. Madson, Niel K., "Convergence of Singular Difference Approximations for the Discrete Ordinate Equations in x-y Geometry," WAPD-T-2382 (October 1970).

39. Madson, Niel K., "Pointwise Convergence of the Discrete Ordinate Method," WAPD-T-2233 (1969).

40. Clifford, C. E., F. R. Mynatt, and E. A. Straker, "Transport Solution to Shielding Problems: Some Recent Developments," Nuclear News (February 1969).

41. Carlson, Bengt G. and G. I. Bell, "Solution of the Transport Equation by the S_N Method," Proceedings Second Intl. Conf. Peaceful Uses of Atomic Energy, 16, 535, United Nations, Geneva (1958).

42. Engle, W. W. Jr., and F. R. Mynatt, "A Comparison at Two Methods of Inner Iteration Convergence Acceleration in Discrete Ordinates Codes," Transactions American Nuclear Society, 11, 193 (1968).

43. Clancy, B. E., and I. J. Donnelly, "Outer Iteration Scaling in Neutron Transport Codes," Nuclear Science and Engineering, 39, 3 (1970).

44. Hageman, L. A., "The Chebyshev Polynomial Method of Iteration," WAPD-TM-537, Westinghouse Electric Corp. (January 1967).

45. Kopp, H. J., "Synthetic Method Solution of the Neutron Transport Equation," Ph.D. Thesis, University of California at Berkeley (1962).

46. Gelbard, Ely M., and L. A. Hageman, "The Synthetic Method as Applied to the S_N Equations," Nuclear Science and Engineering, 37, 288 (1969).

47. Wachspress, E. L., Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics, Prentice Hall, Englewood Cliffs, New Jersey (1966).

48. Lathrop, K. D., "Theory and Use of the General-Geometry TWOTRAN Program," Los Alamos Scientific Laboratory Report LA-4432 (1970).

49. Nakamura, Shoichiro, "A Variational Rebalancing Method for Linear Iterative Convergence Schemes of Neutron Diffusion and Transport Equations," Nuclear Science and Engineering, 39, 2, 278 (1970).

50. Devillers, Christian, "Comparative Performances of Monte Carlo and Discrete Ordinate One Dimensional Transport Codes in an Iron Bulk Shield Calculation," A Review of the Monte Carlo Method for Radiation Transport Calculations, ORNL-RSIC-29 (February 1971).

51. Goldstein, H., et al., "The Role of Cross Section Minima in the Deep Penetration of Fast Neutrons, Proceedings of the Third Conference on Neutron Cross Sections and

Technology, University of Tennessee, Knoxville, Tennessee (August 1971).

52. Reed, Wm. H., "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," Nuclear Science and Engineering, 45, 3, 245 (September 1971).

53. Isaacson, Eugene and Herbert B. Keller, Analysis of Numerical Methods, Wiley & Sons, Inc., New York (1966).

54. Rall, Louis B., Computational Solution of Nonlinear Operator Equations, Wiley & Sons, Inc., New York (1969).

55. Bell, George I., Samuel Glasstone, Nuclear Reactor Theory, Van Nostrand Reinhold Company, New York (1970).

56. Mynatt, F., "Discrete Ordinates S_N Method," Weapons Radiation Shielding Handbook, DASA-1892-3, Chapter 3, p. 13 (1968).

57. Wylie, C. R. Jr., Advanced Engineering Mathematics, Third Edition, p. 430, McGraw-Hill, New York (1966).

58. Engle, W. W. Jr., "A Users Manual for ANISN," USAEC Report K-1693, Union Carbide Corporation (March 1967).

59. Lathrop, K. D., "DTF-IV, A Fortran-IV Program for Solving the Multigroup Transport Equation with Anisotropic Scattering," USAEC Report LA-3373, Los Alamos Scientific Laboratory (November 1965).

60. Lathrop, K. D., "Spatial Differencing of the Transport Equation: Positivity versus Accuracy," Neutron Transport Theory Conference, AEC Report ORO 3858-1 (1969).

61. Abramowitz, Milton and Irene A. Stegun, Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series 55, Seventh Printing (May 1968).

62. Carlson, Bengt G., "Transport Theory: Discrete Ordinates Quadrature over the Unit Sphere," LA-4554, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (September 1970).

63. Clark, Francis H., "The Exponential Transform as an Importance Sampling Device - A Review," ORNL-RSIC-14, Oak Ridge National Laboratory (January 1966).

64. Forsythe, George E. and Cleve B. Moler, "Computer Solution of Linear Algebraic Systems," Prentice-Hall, Inc., Englewood Cliffs, N.J. (1967).

APPENDIX A

FULL-RANGE S_N PROGRAM AND SAMPLE OUTPUT

```
      C  THIS PROGRAM IMPLEMENTS THE DISCRETE ORDINATES ALGORITHM TO SOLVE
      C  THE SLAB PROBLEM WITH A UNIFORM SOURCE.
      C  THE VARIABLES EMPLOYED ARE:
      C    A=WIDTH OF SLAB IN CM.
      C    R=INTEGER GIVING THE NUMBER OF SPATIAL SEGMENTS.
      C    M=NUMBER OF SPACE FLUX POINTS, INCLUDING BOTH BOUNDARIES.
      C    SIGMA=TOTAL COLLISION CROSS SECTION IN UNITS 1/CM.
      C    SIGMAS=MACROSCOPIC SCATTERING CROSS SECTION IN UNITS 1/CM.
      C    WEIGHT(J)=WEIGHTS OF THE QUADRATURE SET.
      C    MU(J)=DISCRETE SET OF DIRECTION COSINES OF ANGLE WITH
      C          RESPECT TO THE X-AXIS.
      C    PHI(K,J)=DIRECTIONAL FLUX AT SPATIAL POINTS X(K) DESCRIBING
      C             NEUTRONS GOING IN DIRECTION MU(J).
      C    MIDSOR(K,J)=TOTAL SOURCE TERM EVALUATED AT SPATIAL MIDPOINTS
      C                AND MU(J).  COMPOSED OF SCATTERING TERM PLUS
      C                THE EXTERNAL SOURCE.
      C    MIDPHI(K,J)=DIRECTIONAL FLUX EVALUATED AT SPATIAL MIDPOINTS
      C                DETERMINED FROM PRESENT ITERATION OF PHI(K,J)
      C                AND USED TO CALCULATE NEXT ITERATE SCATTERING
      C                SOURCE TERM.
      C    SOURCE(K,J)=EXTERNAL SOURCE AT SPATIAL MIDPOINTS AND MU(J).
      C                THIS TERM DOES NOT CHANGE WITH EACH ITERATION.
      C    EPS=CONVERGENCE CRITERION BOUND FOR FRACTIONAL DIFFERENCES.
      C    EPS1=UPPER BOUND ON THE FRACTIONAL ERROR.
      C    NORM=INFINITY NORM OF THE ITERATION MATRIX.
      C    E(J)=MATRIX ELEMENT OF (E-E) CORRESPONDING TO MU(J).
      C    D(J)=MATRIX ELEMENT OF (D-D) CORRESPONDING TO MU(J).
      C
      C
0001        INTEGER R
0002        REAL*8 MU(6),MIDSOR(30,6),MIDPHI(30,6),NORM
0003        DOUBLE PRECISION PHI(31,6),SOURCE(30,6),SIGMA,SIGMAS,
           1WEIGHT(6),A,DELTA,SUM,SUM1,SUM2,X(31),OLDPHI(31,6),
           1CONV,D(6),E(6),EPS,EPS1
0004        COMMON /STORE/PHI,X,MU
0005        EPS1=0.0001
0006        A=10.0
0007        R=30
0008        M=R+1
0009        SIGMA=1.0
0010        SIGMAS=0.5
0011        MU(1) = .23862
0012        MU(2) = .66121
0013        MU(3) = .93247
0014        MU(4) = -.23862
0015        MU(5) = -.66121
0016        MU(6) = -.93247
0017        WEIGHT(1) = .46791
0018        WEIGHT(2) = .36076
0019        WEIGHT(3) = .17132
0020        WEIGHT(4) = .46791
0021        WEIGHT(5) = .36076
0022        WEIGHT(6) = .17132
0023        DELTA=A/R
      C  THIS PART OF THE PROGRAM CALCULATES THE ALGORITHM COEFFICIENTS.
```

```
      DO 30 I=1,3
      E(I)=(2.*MU(I)-DELTA*SIGMA)/(2.*DELTA)
      D(I)=(2.*MU(I)+DELTA*SIGMA)/(2.*DELTA)
      J=3+I
      E(J)=E(I)
   30 D(J)=D(I)
C  THIS PART OF THE PROGRAM COMPUTES THE NORM OF THE ITERATION
C  MATRIX.  TO BE VALID MU(1) MUST BE THE MINIMUM ABSOLUTE
C  VALUE OF THE COMPONENTS OF THE DIRECTION SET.
      J=M-3
      SUM=E(1)/D(1)
      SUM1=1.
      SUM2=0.
      DO 40 I=1,J
      SUM1=SUM1*SUM
   40 SUM2=SUM2+SUM1
      SUM1=SUM1*SUM
      NORM=SIGMAS/4.*(4./D(1)*(1.+SUM2)+3./D(1)*SUM1)
C  THIS PART OF PROGRAM CALCULATES THE ALGORITHM CONVERGENCE
C  CRITERION.
      EPS=EPS1/(1.-EPS1)*(1.-NORM)
      DO 50 I=1,M
   50 X(I)=(I-1)*DELTA
      DO 80 J=1,6
      DO 80 K=1,M
      SOURCE(K,J)=1.0
   80 MIDSCE(K,J)=1.0
      DO 85 J=1,6
      DO 85 K=1,M
   85 PHI(K,J)=0.
C  THIS PART OF THE PROGRAM STARTS THE INNER ITERATION.  NOTE
C  THAT THE INITIAL FLUX VECTOR GUESS IS THE ZERO VECTOR.
      DO 312 INNER=1,1000
      DO 175 J=1,3
      DO 150 K=2,M
      IF(K .EQ. 2) PHI(2,J)=1./D(J)*MIDSCE(1,J)
      IF (K .EQ. 2) GO TO 150
      PHI(K,J)=E(J)/D(J)*PHI(K-1,J)+1./D(J)*MIDSCE(K-1,J)
  150 CONTINUE
  175 CONTINUE
      DO 190 J=4,6
      DO 185 K=1, M
      N = M-K
      IF(N .EQ. (M-1)) PHI(N,J)=1./D(J)*MIDSCE(N,J)
      IF (N .EQ. (M-1) ) GO TO 185
      PHI(N,J)=E(J)/D(J)*PHI(N+1,J) + 1./D(J)*MIDSCE(N,J)
  185 CONTINUE
  190 CONTINUE
C    THIS LOOP COMPUTES THE NEW SPATIAL MID-POINT FLUXES FROM THE PRESENT
C    ITERATE FLUXES.
      DO 230 J=1,6
      DO 230 K=1,M
  230 MIDPHI(K,J) = ( PHI(K+1,J) + PHI(K,J))/2.
C  CHECK FOR POINTWISE CONVERGENCE IS CONDUCTED.
C  NO CONVERGENCE CHECK IS MADE THE FIRST TIME THROUGH.
```

```
      IF (INNER .EQ. 1) GO TO 280
C  A TABLE SEARCH IS CONDUCTED FOR THE MAXIMUM DIFFERENCE.
      SUM=0.
      DO 260 K=1,M
      DO 260 J=1,6
      SUM1=PHI(K,J)-OLDPHI(K,J)
      IF(SUM1 .GT. SUM) SUM=SUM1
  260 CONTINUE
      DO 265 K=2,M
      DO 265 J=1,3
      CONV=SUM/OLDPHI(K,J)
      IF(CONV .GT. EPS) GO TO 280
  265 CONTINUE
      DO 270 K=1,M
      DO 270 J=4,6
      CONV=SUM/OLDPHI(K,J)
      IF(CONV .GT. EPS) GO TO 280
  270 CONTINUE
      GO TO 600
  280 CONTINUE
C  THIS LOOP CALCULATES THE NEW SOURCE.
      DO 290 K=1,M
      DO 290 J=1,6
      SUM1=0.
      DO 285 I=1,6
  285 SUM1=SUM1+WEIGHT(I)*SIDPHI(K,I)
  290 MIDSOR(K,J)=SIGMAS/2. * SUM1 + SOURCE(K,J)
      DO 295 K=1,M
      DO 295 J=1,6
  295 OLDPHI(K,J)=PHI(K,J)
  312 CONTINUE
  600 WRITE(6,610)
  610 FORMAT('1',5X,'THE UNMODIFIED DISCRETE ORDINATES ',
     1'ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING',
     1' PROBLEM PARAMETERS')
      WRITE(6,620) A,SIGMA,SIGMAS,M
  620 FORMAT('0',19X,'SLABWIDTH=',F4.1,' CM.'/20X,'TOTAL ',
     1'MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,'SCATTERI',
     1'NG MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,
     1'NUMBER OF SPATIAL MESH INCREMENTS=',I3,' .')
      WRITE(6,625) DELTA
  625 FORMAT(' ',19X,'SIZE OF EACH SPATIAL MESH INCREMENT=',
     1F7.5,' CM.'/20X,'ANGULAR QUADRATURE SET USED=SIX ',
     1'ORDINATE GAUSS-LEGENDRE.'/20X,'SOURCE=UNIFORM',
     1' WITH UNIT STRENGTH.')
      WRITE(6,630) EPS,INNER,EPS1
  630 FORMAT('0',4X,'THE FRACTIONAL DIFFERENCE CONVERGENCE',
     1' BOUND,',E12.5,', WAS MET AFTER ',I5,' ITERATIONS.',
     1'  THIS IMPLIES '/5X,'THAT THE FRACTIONAL ITERATION',
     1' ERROR IS LESS THAN ',F6.4,' .')
      SUM=E(1)/D(1)
      SUM1=1./D(1)
      WRITE(6,640) NORM,SUM,SUM1
  640 FORMAT('0',4X,'THE INFINITY NORM OF THE ITERATION',
     1' MATRIX=',F7.5,' BASED ON THE VALUES OF E(1)/D(1)=',
```

```
     1F7.5,' AND 1/D(1)=',F7.5,' .')
      WRITE(6,650)
  650 FORMAT('0',30X,'THE CONVERGED ITERATE ANGULAR FLUXES ',
     1'ARE LISTED')
      WRITE(6,660)
  660 FORMAT(' ',10X,'X,CM.',7X,'MU(1)',9X,'MU(2)',9X,'MU(3)',
     19X,'MU(4)',9X,'MU(5)',9X,'MU(6)')
      DO 665 K=1,M
  665 WRITE(6,670) X(K),(PHI(K,J),J=1,6)
  670 FORMAT(' ',8X,F7.4,6(2X,E12.5))
      WRITE(6,680)
  680 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
      WRITE(6,690)
  690 FORMAT('1',10X,'THE PREVIOUS ITERATE (BEFORE CONVER',
     1'GENCE) ANGULAR FLUXES ARE LISTED')
      WRITE(6,700)
  700 FORMAT(' ',10X,'X,CM.',7X,'MU(1)',9X,'MU(2)',9X,'MU(3)',
     19X,'MU(4)',9X,'MU(5)',9X,'MU(6)')
      DO 705 K=1,M
  705 WRITE(6,710) X(K),(OLDPHI(K,J),J=1,6)
  710 FORMAT(' ',8X,F7.4,6(2X,E12.5))
      WRITE(6,720)
  720 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
      DO 900 J=1,6
  900 CALL PLOT(M,J)
      END
```

THE UNMODIFIED DISCRETE ORDINATES ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING PROBLEM PARAMETERS

SLABWIDTH=10.0 CM.
TOTAL MACROSCOPIC CROSS SECTION=1.0000 1/CM.
SCATTERING MACROSCOPIC CROSS SECTION=0.9999 1/CM.
NUMBER OF SPATIAL MESH INCREMENTS= 30 .
SIZE OF EACH SPATIAL MESH INCREMENT=0.33333 CM.
ANGULAR QUADRATURE SET USED=SIX ORDINATE GAUSS-LEGENDRE.
SOURCE=UNIFORM WITH UNIT STRENGTH.

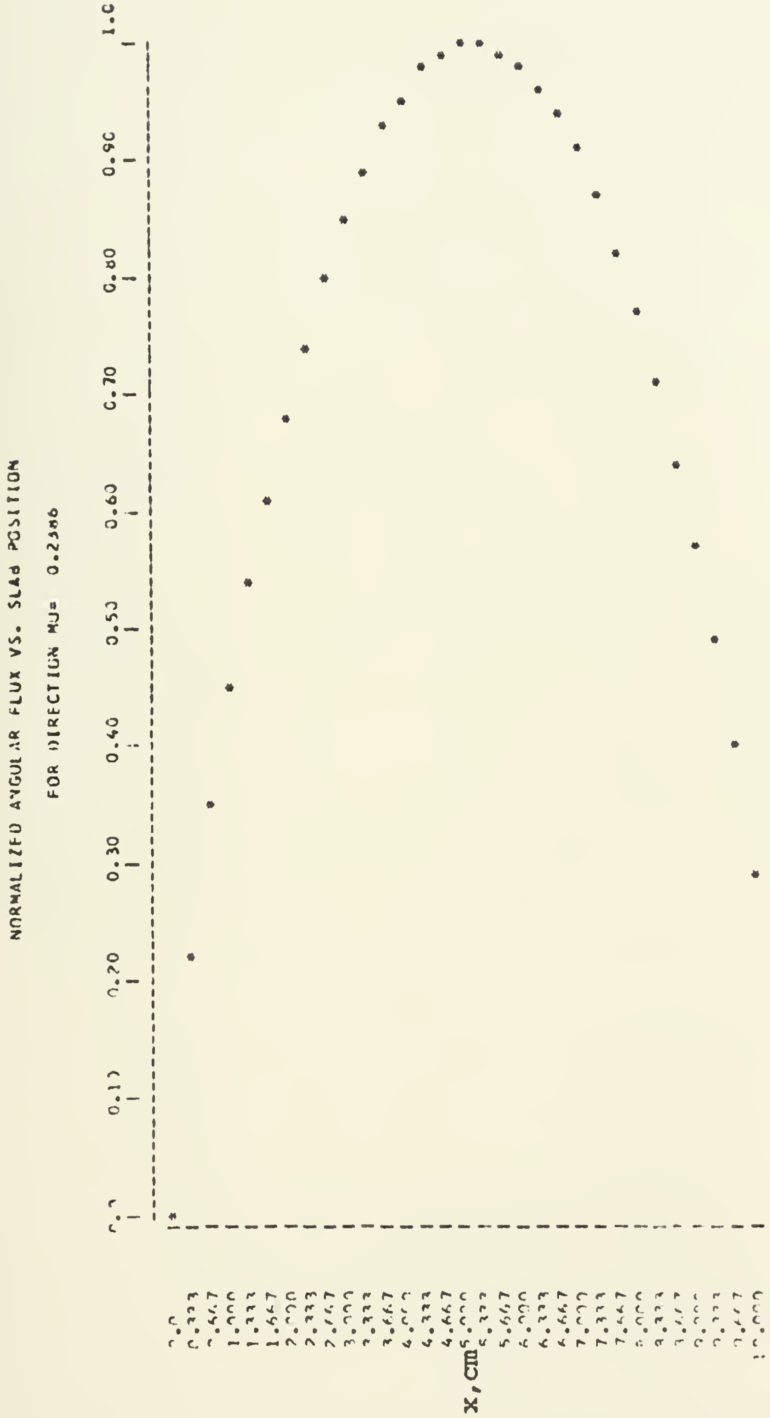
THE FRACTIONAL DIFFERENCE CONVERGENCE BOUND, 0.10000D-07, WAS MET AFTER 699 ITERATIONS. THIS IMPLIES THAT THE FRACTIONAL ITERATION ERROR IS LESS THAN 0.0001 .

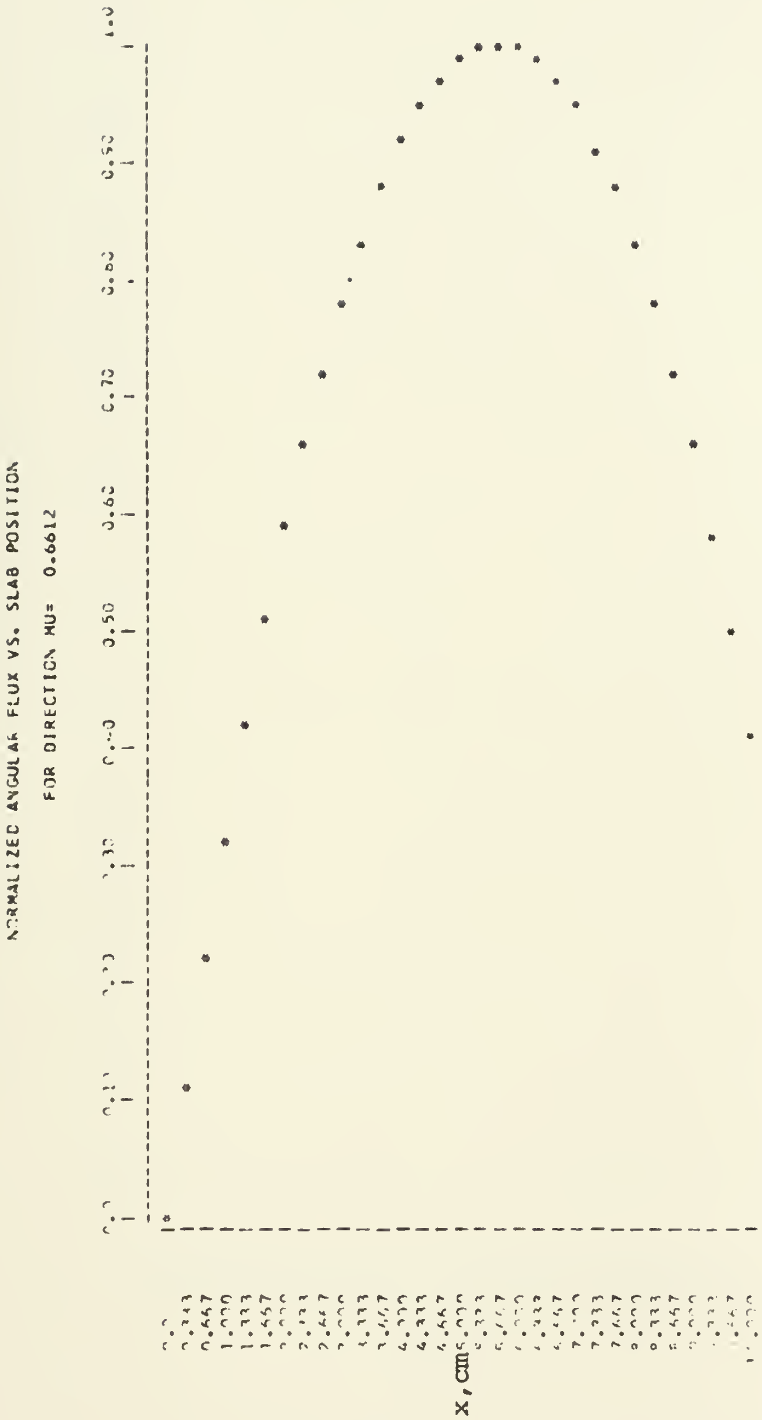
THE INFINITY NORM OF THE ITERATION MATRIX=0.99990 BASED ON THE VALUES OF E(1)/D(1)=0.17754 AND 1/D(1)=0.82246 .

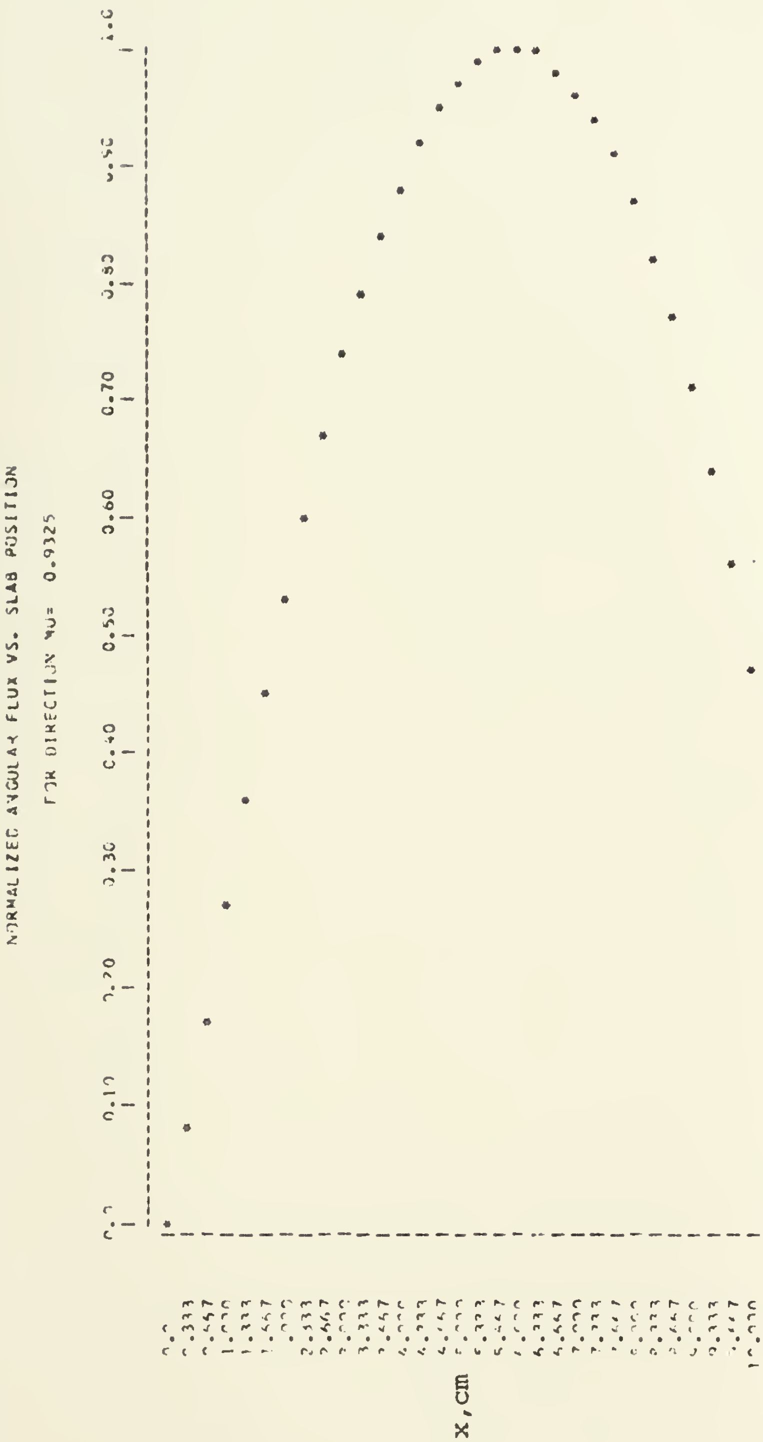
THE CONVERGED ITERATE ANGULAR FLUXES ARE LISTED

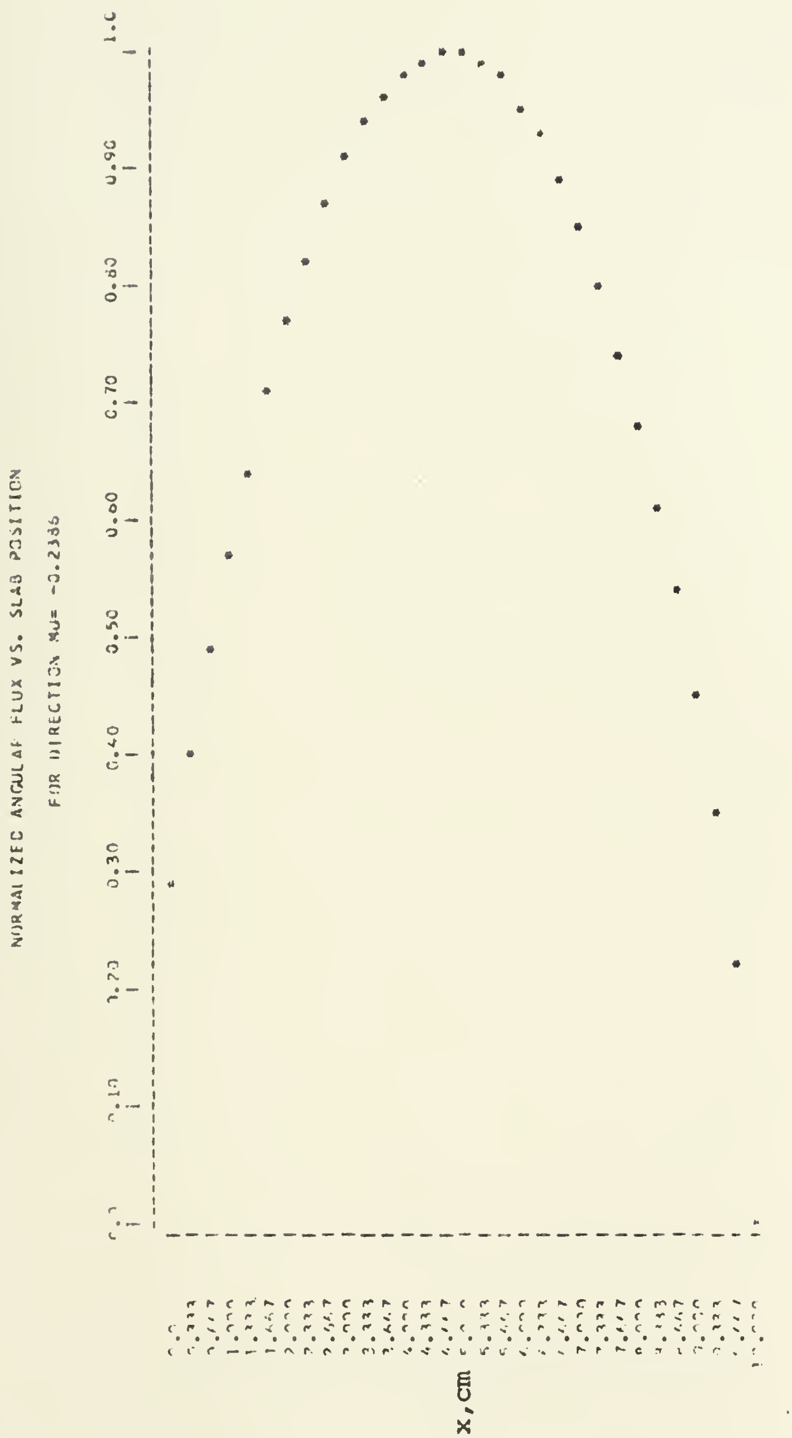
X,CM.	MU(1)	MU(2)	MU(3)	MU(4)	MU(5)	MU(6)
0.0	0.0	0.0	0.0	0.14133D 02	0.19870D 02	0.22799D 02
0.3333	0.10566D 02	0.52213D 01	0.39327D 01	0.19561D 02	0.24522D 02	0.27078D 02
0.6667	0.17175D 02	0.10600D 02	0.83747D 01	0.24107D 02	0.28528D 02	0.30777D 02
1.0000	0.22160D 02	0.15688D 02	0.12882D 02	0.28140D 02	0.32095D 02	0.34059D 02
1.3333	0.26433D 02	0.20385D 02	0.17271D 02	0.31775D 02	0.35289D 02	0.36977D 02
1.6667	0.30245D 02	0.24686D 02	0.21455D 02	0.35051D 02	0.38135D 02	0.39549D 02
2.0000	0.33677D 02	0.28605D 02	0.25386D 02	0.37982D 02	0.40640D 02	0.41782D 02
2.3333	0.36758D 02	0.32155D 02	0.29036D 02	0.40575D 02	0.42809D 02	0.43680D 02
2.6667	0.39496D 02	0.35349D 02	0.32386D 02	0.42832D 02	0.44643D 02	0.45244D 02
3.0000	0.41898D 02	0.38195D 02	0.35429D 02	0.44754D 02	0.46144D 02	0.46475D 02
3.3333	0.43965D 02	0.40698D 02	0.38153D 02	0.46343D 02	0.47312D 02	0.47373D 02
3.6667	0.45698D 02	0.42862D 02	0.40555D 02	0.47600D 02	0.48147D 02	0.47939D 02
4.0000	0.47298D 02	0.44689D 02	0.42631D 02	0.48524D 02	0.48650D 02	0.48173D 02
4.3333	0.48166D 02	0.46182D 02	0.44379D 02	0.49116D 02	0.48821D 02	0.48076D 02
4.6667	0.48901D 02	0.47341D 02	0.45799D 02	0.49376D 02	0.48660D 02	0.47648D 02
5.0000	0.49305D 02	0.48167D 02	0.46888D 02	0.49305D 02	0.48167D 02	0.46888D 02
5.3333	0.49376D 02	0.48660D 02	0.47648D 02	0.48901D 02	0.47341D 02	0.45799D 02
5.6667	0.49116D 02	0.48821D 02	0.48076D 02	0.48166D 02	0.46182D 02	0.44379D 02
6.0000	0.48524D 02	0.48650D 02	0.48173D 02	0.47298D 02	0.44689D 02	0.42631D 02
6.3333	0.47600D 02	0.48147D 02	0.47939D 02	0.45698D 02	0.42862D 02	0.40555D 02
6.6667	0.46343D 02	0.47312D 02	0.47373D 02	0.43965D 02	0.40698D 02	0.38153D 02
7.0000	0.44754D 02	0.46144D 02	0.46475D 02	0.41898D 02	0.38195D 02	0.35429D 02
7.3333	0.42832D 02	0.44643D 02	0.45244D 02	0.39496D 02	0.35349D 02	0.32386D 02
7.6667	0.40575D 02	0.42809D 02	0.43680D 02	0.36758D 02	0.32155D 02	0.29036D 02
8.0000	0.37982D 02	0.40640D 02	0.41782D 02	0.33677D 02	0.28605D 02	0.25386D 02
8.3333	0.35051D 02	0.38135D 02	0.39549D 02	0.30245D 02	0.24686D 02	0.21455D 02
8.6667	0.31775D 02	0.35289D 02	0.36977D 02	0.26433D 02	0.20385D 02	0.17271D 02
9.0000	0.28140D 02	0.32095D 02	0.34059D 02	0.22160D 02	0.15688D 02	0.12882D 02
9.3333	0.24107D 02	0.28528D 02	0.30777D 02	0.17175D 02	0.10600D 02	0.83747D 01
9.6667	0.19561D 02	0.24522D 02	0.27078D 02	0.10566D 02	0.52213D 01	0.39327D 01
10.0000	0.14133D 02	0.19870D 02	0.22799D 02	0.0	0.0	0.0

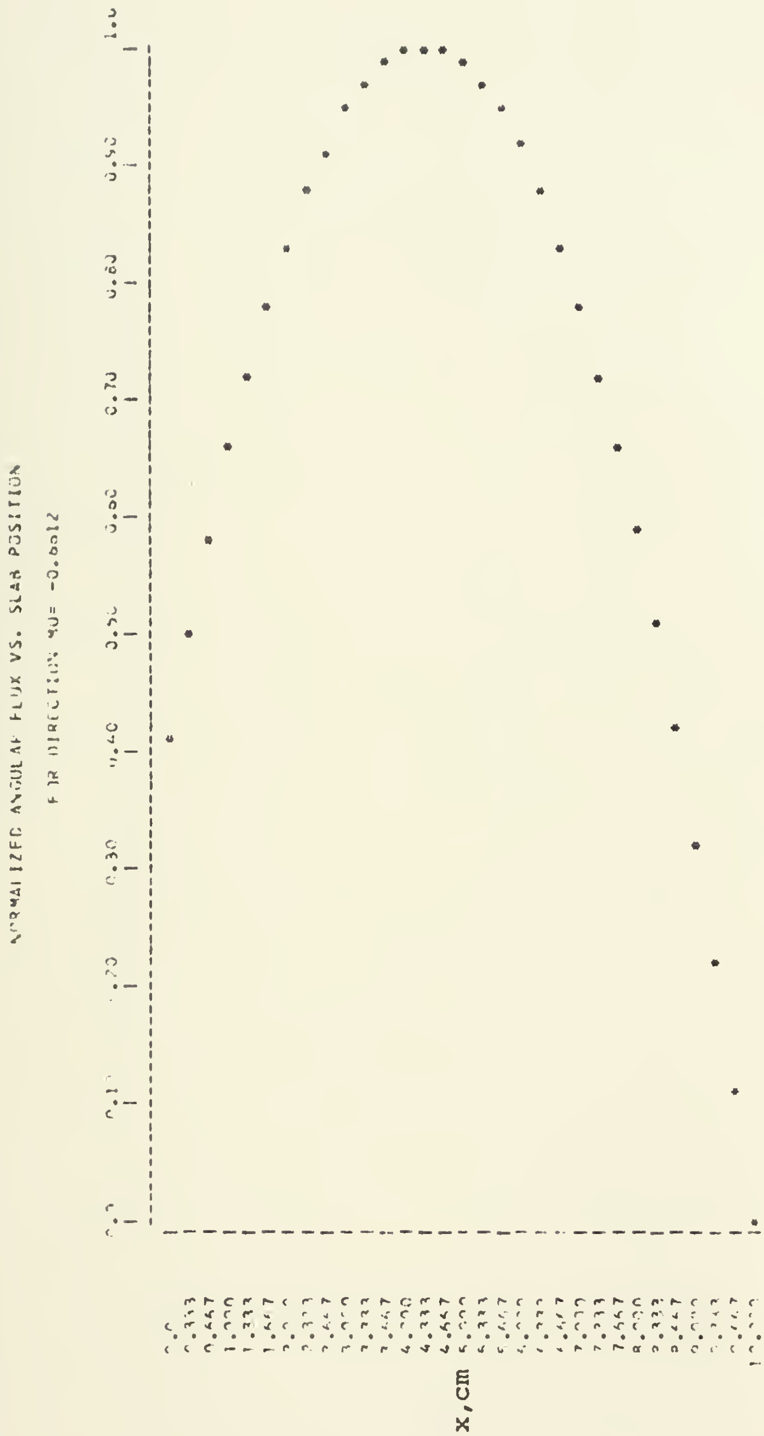
NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

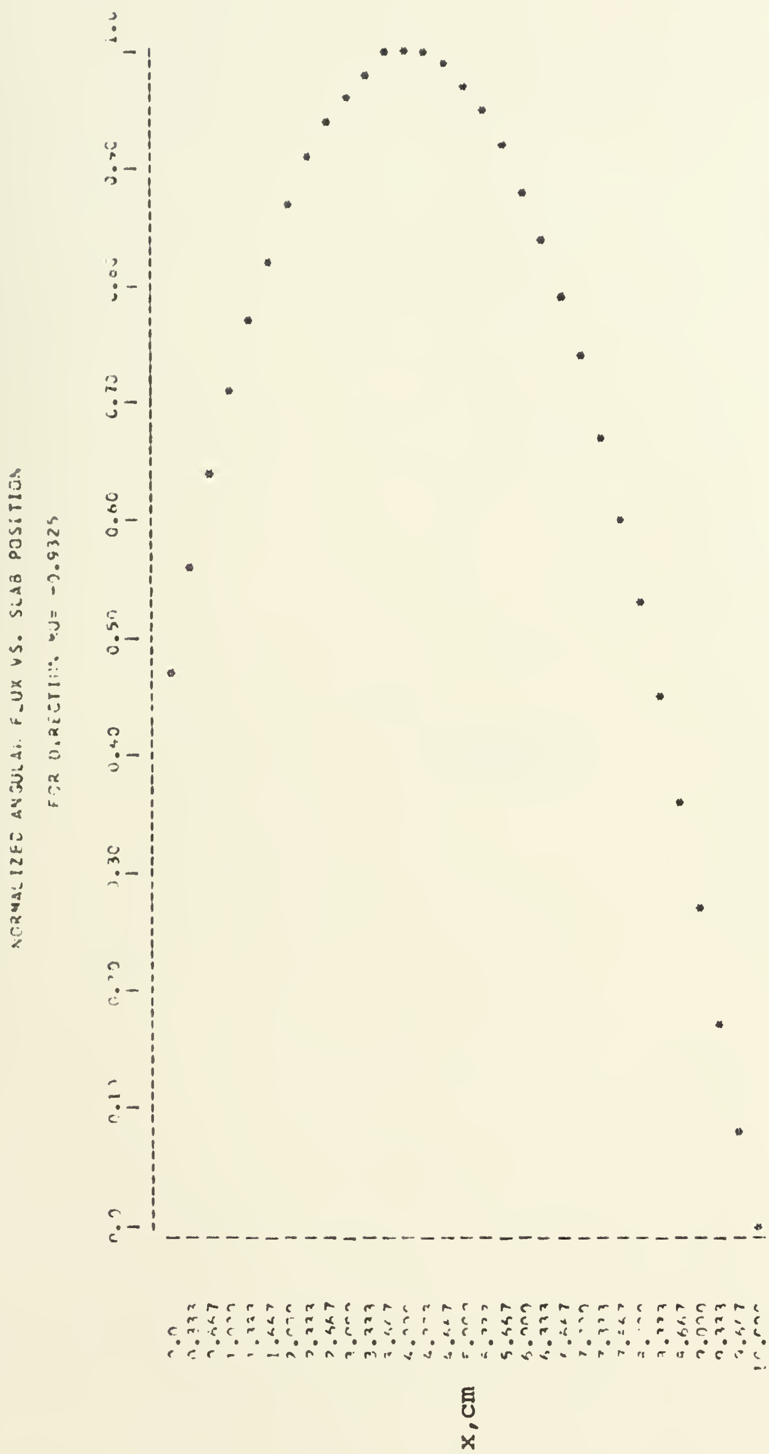

NORMALIZED ANGULAR FLUX VS. SLAB POSITION
FOR DIRECTION MU= 0.2386
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.0
x,cm


NORMALIZED ANGULAR FLUX VS. SLAB POSITION
FOR DIRECTION MU= 0.6612
x,cms


NORMALIZED ANGULAR FLUX VS. SLAB POSITION
FOR DIRECTION MU= 0.9325
x,cm


NORMALIZED ANGULAR FLUX VS. SLAB POSITION
FOR DIRECTION MU= -0.2386
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.0
x,cm


NORMALIZED ANGULAR FLUX VS. SLAB POSITION
FOR DIRECTION MU= -0.6612
0.10
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.0
x,cm


NORMALIZED ANGULAR FLUX VS. SLAB POSITION
FOR DIRECTION MU= -0.9325
x,cm

APPENDIX B

APPENDIX B

Properties of $||(D-E)^{-1}S||$

The norm of the transform domain iteration matrix is described by

$$||(D-E)^{-1}S|| = \frac{\sigma^s}{2d_i}\left[2\left(\frac{1-\gamma^R}{1-\gamma}\right)-\gamma^{R-1}\right] \tag{1}$$

in which

$$0 \leq \gamma = \frac{|e_i|}{d_i} < 1 \tag{2}$$

and

$$e_i = \frac{2(\mu_i)_{min}-\Delta\sigma^t-\Delta\alpha(\mu_i)_{min}}{2\Delta} \tag{3}$$

$$d_i = \frac{2(\mu_i)_{min}+\Delta\sigma^t+\Delta\alpha(\mu_i)_{min}}{2\Delta} \tag{4}$$

and R is the number of spatial intervals in the mesh.

The fact that $||(D-E)^{-1}S||$ is a monotone decreasing function of α is verified by analysis of three regions of α; small α near zero, at $\alpha = \alpha^*$, and for $\alpha > \alpha^*$ where

$$\alpha^* = \frac{2(\mu_i)_{min}-\Delta\sigma^t}{2(\mu_i)_{min}} . \tag{5}$$

The derivative of $||(D-E)^{-1}S||$ with respect to α is

$$\frac{d||(D-E)^{-1}S||}{d\alpha} = -\frac{\sigma^s \mu_i}{2d_i^2}\left[2\left(\frac{1-\gamma^R}{1-\gamma}\right)-\gamma^{R-1}\right] +$$

$$+\begin{cases} -\dfrac{\sigma^s\mu_i^2}{2\Delta d_i^3}\left\{2\left[\dfrac{(1-\gamma^R)-R\gamma^{R-1}(1-\gamma)}{(1-\gamma)^2}\right]-(R-1)\gamma^{R-2}\right\} & \text{for } 2\mu_i-\Delta\sigma^t-\Delta\alpha\mu_i>0. \quad (6a) \\ 0 & \text{for } 2\mu_i-\Delta\sigma^t-\Delta\alpha\mu_i=0. \quad (6b) \end{cases}$$

The relationships

$$\frac{d(d_i)}{d\alpha} = \frac{\mu_i}{2} \tag{7}$$

$$\frac{d\gamma}{d\alpha} = -\frac{\mu_i^2}{\Delta d_i^2} \tag{8}$$

for $2\mu_i-\Delta\sigma^t-\Delta\alpha\mu_i > 0$ have been used in the above determinations.

When α is small and $2\mu_i-\Delta\sigma^t-\Delta\alpha\mu_i > 0$,

$$\gamma = \frac{|e_i|}{d_i} = \frac{2\mu_i-\Delta\sigma^t-\Delta\alpha\mu_i}{2\mu_i+\Delta\sigma^t+\Delta\alpha\mu_i} = 1-\varepsilon \tag{9}$$

where

$$\varepsilon = \frac{\Delta\left(\alpha+\dfrac{\sigma^t}{\mu_i}\right)}{\left(1+\dfrac{\Delta\sigma^t}{2\mu_i}+\dfrac{\Delta\alpha}{2}\right)} . \tag{10}$$

Expanding for small ε,

$$\gamma^R = (1-\varepsilon)^R \simeq 1-R\varepsilon \tag{11}$$

and substituting into (6a) produces

$$\frac{d||(D-E)^{-1}S||}{d\alpha} \simeq -\frac{\sigma^s\mu_i}{2d_i^2}\Big[(2R-1)+\varepsilon(R-1)\Big] - \frac{\sigma^s\mu_i^2}{2\Delta d_i^3}(R-1)\Big[R(2+\varepsilon)-(1+\varepsilon)\Big] \ . \tag{12}$$

This quantity is negative for all R. When R is large, as in most practical problems, $||(D-E)^{-1}S||$ has a large negative slope.

When $\alpha = \alpha^*$, $2\mu_i-\Delta\sigma^t-\Delta\alpha\mu_i = 0$ and $e_i = \gamma = 0$. In this case

$$\frac{d||(D-E)^{-1}S||}{d\alpha} = -\frac{\sigma^s\mu_i}{d_i^2} \tag{13}$$

from (6b). Observe that the slope of $||(D-E)^{-1}S||$ is still negative but is much smaller in magnitude than when α is small near zero.

By definition (5), α^* is the value of α when $e_i = 0$, that is,

$$\Delta\alpha^*(\mu_i)_{min}+\Delta\sigma^t-2(\mu_i)_{min} = 0 \ . \tag{14}$$

So at α^*, $\gamma = 0$ and

$$||(D-E)^{-1}S|| = \frac{\sigma^s}{d_i} = \frac{\Delta\sigma^s}{2(\mu_i)_{min}} \ . \tag{15}$$

For $\alpha > \alpha^*$, let

$$\alpha = \alpha^* + \varepsilon \tag{16}$$

where $0 < \varepsilon < \infty$. Then

$$\gamma = \frac{\Delta\varepsilon}{4+\Delta\varepsilon} \tag{17}$$

$$\frac{1}{d_i} = \frac{2\Delta}{(\mu_i)_{min}}\left(\frac{1}{4+\Delta\varepsilon}\right) \tag{18}$$

and

$$||(D-E)^{-1}s|| = \frac{\Delta\sigma^s}{2(\mu_i)_{min}}\left\{1 - \frac{(\Delta\varepsilon)^R + 2(\Delta\varepsilon)^{R-1}}{(4+\Delta\varepsilon)^R}\right\} . \tag{19}$$

The derivative of $||(D-E)^{-1}s||$ with respect to $\Delta\varepsilon$ is,

$$\frac{d||(D-E)^{-1}s||}{d(\Delta\varepsilon)} = - \frac{2(2R-1)(\Delta\varepsilon)^{R-1} + 8(R-1)(\Delta\varepsilon)^{R-2}}{(4+\Delta\varepsilon)^{R+1}} \tag{20}$$

which is negative for all ε, $0 < \varepsilon < \infty$.

Equations (12), (13), and (20) reveal that the norm of the transform domain iteration has a negative slope for $0 < \alpha < \infty$, consequently it is a monotone decreasing function of α.

Equation (19) displays that

$$\lim_{\alpha\to\infty}||(D-E)^{-1}s|| = \lim_{\Delta\varepsilon\to\infty}\left\{\frac{\Delta\sigma^s}{2(\mu_i)_{min}}\left[1-\frac{(\Delta\varepsilon)^R + 2(\Delta\varepsilon)^{R-1}}{(4+\Delta\varepsilon)^R}\right]\right\}$$

$$= 0$$

since

$$\lim_{\Delta\varepsilon\to\infty}\left[\frac{(\Delta\varepsilon)^R + 2(\Delta\varepsilon)^{R-1}}{(4+\Delta\varepsilon)^R}\right] = \frac{R!}{R!} = 1$$

by R applications of ℓ'Hospital's rule.

APPENDIX C

HALF-RANGE UNMODIFIED S_N PROGRAM AND SAMPLE OUTPUT

```
C  THIS PROGRAM IMPLEMENTS THE UNMODIFIED DISCRETE ORDINATES ALGORITHM
C  TO SOLVE THE CASE I HALF-RANGE SLAB PROBLEM WITH FIRST COLLISION
C  SOURCE.
C  THE VARIABLES EMPLOYED ARE:
C    A=WIDTH OF SLAB IN CM.
C    R=INTEGER GIVING THE NUMBER OF SPATIAL SEGMENTS.
C    M=NUMBER OF SPACE FLUX POINTS, INCLUDING BOTH BOUNDARIES.
C    SIGMA=TOTAL COLLISION CROSS SECTION IN UNITS 1/CM.
C    SIGMAS=MACROSCOPIC SCATTERING CROSS SECTION IN UNITS 1/CM.
C    WEIGHT(J)=WEIGHTS OF THE QUADRATURE SET.
C    MU(J)=DISCRETE SET OF DIRECTION COSINES OF ANGLE WITH
C          RESPECT TO THE X-AXIS.
C    PHI(K,J)=DIRECTIONAL FLUX AT SPATIAL POINTS X(K) DESCRIBING
C          NEUTRONS GOING IN DIRECTION MU(J).
C    MIDSOR(K,J)=TOTAL SOURCE TERM EVALUATED AT SPATIAL MIDPOINTS
C                AND MU(J).  COMPOSED OF SCATTERING TERM PLUS
C                THE EXTERNAL SOURCE.
C    MIDPHI(K,J)=DIRECTIONAL FLUX EVALUATED AT SPATIAL MIDPOINTS
C                DETERMINED FROM PRESENT ITERATION OF PHI(K,J)
C                AND USED TO CALCULATE NEXT ITERATE SCATTERING
C                SOURCE TERM.
C    SOURCE(K,J)=EXTERNAL SOURCE AT SPATIAL MIDPOINTS AND MU(J).
C                THIS TERM DOES NOT CHANGE WITH EACH ITERATION.
C    EPS=A CONVERGENCE CRITERION BOUND FOR FRACTIONAL DIFFERENCES.
C    EPS1=UPPER BOUND ON THE FRACTIONAL ERROR.
C    NORM=INFINITY NORM OF THE ITERATION MATRIX.
C    E(J)=MATRIX ELEMENT OF (D-E) CORRESPONDING TO MU(J).
C    D(J)=MATRIX ELEMENT OF (D-E) CORRESPONDING TO MU(J).
C
C
      INTEGER R
      REAL*8 MU(3),MIDSOR(30,3),MIDPHI(30,3),NORM
      DOUBLE PRECISION PHI(31,3),SOURCE(30,3),SIGMA,SIGMAS,
     1WEIGHT(3),A,DELTA,SUM,SUM1,SUM2,X(31),OLDPHI(31,3),
     1CONV,D(3),E(3),EPS,EPS1
      COMMON /STORE/PHI,X,MU
      EPS1=0.0001
      A=10.0
      R=30
      M=R+1
      SIGMA=1.0
      SIGMAS=0.9999
      MU(1) = .23862
      MU(2) = .66121
      MU(3) = .93247
      WEIGHT(1) = .46791
      WEIGHT(2) = .36076
      WEIGHT(3) = .17132
      DELTA=A/R
C  THIS PART OF THE PROGRAM CALCULATES THE ALGORITHM COEFFICIENTS.
      DO 30 I=1,3
      E(I)=(2.*MU(I)-DELTA*SIGMA)/(2.*DELTA)
   30 D(I)=(2.*MU(I)+DELTA*SIGMA)/(2.*DELTA)
C  THIS PART OF THE PROGRAM COMPUTES THE NORM OF THE ITERATION
C  MATRIX.  IT DEPENDS UPON MU(1) BEING THE MINIMUM IN THE SET.
```

```
      J=P-2
      SUM=DABS(F(1)/D(1))
      SUM1=1.
      SUM2=0.
      DO 45 I=1,J
      SUM1=SUM1*SUM
   45 SUM2=SUM2+SUM1
      SUM1=SUM1*SUM
      NORM=SIGMAS/2.*(2./D(1)*(1.+SUM2)+1./D(1)*SUM1)
C  THIS PART OF PROGRAM CALCULATES THE ALGORITHM CONVERGENCE
C  CRITERION WHICH GUARANTEES THAT FRACTIONAL ERROR IS LESS
C  THAN EPS1.
      EPS=EPS1/(1.-EPS1)*(1.-NORM)
      DO 50 I=1,M
   50 X(I)=(I-1)*DELTA
C  THE FIRST COLLISION SOURCE IS CALCULATED.
      DO 80 J=1,3
      DO 80 K=1,R
      SOURCE(K,J)=DEXP(-SIGMA*(X(K+1)+X(K))/(2.*MU(J)))
   80 MIDSCR(K,J)=SOURCE(K,J)
      DO 95 J=1,3
      DO 95 K=1,M
   95 PHI(K,J)=0.
C  THIS PART OF THE PROGRAM STARTS THE INNER ITERATION.  NOTE
C  THAT THE INITIAL FLUX VECTOR GUESS IS THE ZERO VECTOR.
      DO 312 INNER=1,1000
      DO 175 J=1,3
      DO 150 K=2,M
      PHI(K,J)=F(J)/D(J)*PHI(K-1,J)+1./D(J)*MIDSCR(K-1,J)
150   CONTINUE
175   CONTINUE
C     THIS LOOP COMPUTES THE NEW SPATIAL MID-POINT FLUXES FROM THE PRESENT
C     ITERATE FLUXES.
      DO 230 J=1,3
      DO 230 K=1,R
  230 MIDPHI(K,J) = ( PHI(K+1,J) + PHI(K,J))/2.
C  CHECK FOR POINTWISE CONVERGENCE IS CONDUCTED.
C  NO CONVERGENCE CHECK IS MADE THE FIRST TIME THROUGH.
      IF (INNER .EQ. 1) GO TO 280
C  THE NEW CONVERGENCE CRITERION, BASED ON THE MAXIMUM DIFFERENCE
C  AND ITERATION MATRIX NORM, IS USED.
C  A TABLE SEARCH IS CONDUCTED FOR THE MAXIMUM DIFFERENCE.
      SUM=0.
      DO 255 K=1,M
      DO 255 J=1,3
      SUM1=PHI(K,J)-OLDPHI(K,J)
      IF(SUM1 .GT. SUM) SUM=SUM1
  255 CONTINUE
      DO 260 K=2,M
      DO 260 J=1,3
      CONV=SUM/OLDPHI(K,J)
      IF(CONV .GT. EPS) GO TO 280
  260 CONTINUE
      GO TO 600
  280 CONTINUE
```

```
C   THIS LOOP CALCULATES THE NEW SOURCE.
          DO 290 K=1,R
          DO 290 J=1,3
          SUM1=0.
          DO 285 I=1,3
      285 SUM1=SUM1+WEIGHT(I)*MIDPHI(K,I)
      290 MIDSOR(K,J)=SIGMAS*SUM1+SOURCE(K,J)
          DO 295 K=1,M
          DO 295 J=1,3
      295 OLDPHI(K,J)=PHI(K,J)
      312 CONTINUE
      600 WRITE(6,610)
      610 FORMAT('1',5X,'THE UNMODIFIED DISCRETE ORDINATES ',
         1'ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING',
         1' PROBLEM PARAMETERS')
          WRITE(6,615)
      615 FORMAT('0',19X,'HALF-RANGE CASE I PROBLEM.')
          WRITE(6,620) A,SIGMA,SIGMAS,R
      620 FORMAT(' ',19X,'SLAB WIDTH=',F4.1,' CM.'/20X,'TOTAL ',
         1'MACROSCOPIC CROSS SECTION=',F5.4,' 1/CM.'/20X,'SCATTERI',
         1'NG MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,
         1'NUMBER OF SPATIAL MESH INCREMENTS=',I3,' .')
          WRITE(6,625) DELTA
      625 FORMAT(' ',19X,'SIZE OF EACH SPATIAL MESH INCREMENT=',
         1F7.5,' CM.'/20X,'ANGULAR QUADRATURE SET USED=SIX ',
         1'ORDINATE GAUSS.'/20X,'SOURCE=FIRST COLLISION',
         1' WITH UNIT STRENGTH.')
          WRITE(6,630) EPS,INNER,EPS1
      630 FORMAT('0',4X,'THE FRACTIONAL DIFFERENCE CONVERGENCE',
         1' BOUND,',E12.5,', WAS MET AFTER ',I5,' ITERATIONS.',
         1' THIS IMPLIES '/5X,'THAT THE FRACTIONAL ITERATION',
         1' ERROR IS LESS THAN ',F6.4,' .')
          SUM=E(1)/D(1)
          SUM1=1./D(1)
          WRITE(6,640) NORM,SUM,SUM1
      640 FORMAT('0',4X,'THE INFINITY NORM OF THE ITERATION',
         1' MATRIX=',F7.5,' BASED ON THE VALUES OF E(1)/D(1)=',
         1F7.5,' AND 1/D(1)=',F7.5,' .')
          WRITE(6,650)
      650 FORMAT('0',30X,'THE CONVERGED ITERATE ANGULAR FLUXES ',
         1'ARE LISTED')
          WRITE(6,660)
      660 FORMAT(' ',32X,'X,CM.',11X,'MU(1)',15X,'MU(2)',
         115X,'MU(3)')
          DO 665 K=1,M
      665 WRITE(6,670) X(K),(PHI(K,J),J=1,3)
      670 FORMAT(' ',30X,F7.4,3(8X,E12.5))
          WRITE(6,680)
      680 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
         1' ONE.')
          WRITE(6,690)
      690 FORMAT('1',10X,'THE PREVIOUS ITERATE (BEFORE CONVER',
         1'GENCE) ANGULAR FLUXES ARE LISTED')
          WRITE(6,700)
      700 FORMAT(' ',32X,'X,CM.',11X,'MU(1)',15X,'MU(2)',
```

```
     115X,'MU(3)')
      DO 705 K=1,M
705   WRITE(6,710) X(K),(OLDPHI(K,J),J=1,3)
710   FORMAT(' ',30X,F7.4,3(5X,F12.5))
      WRITE(6,720)
720   FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
      DO 900 J=1,3
900   CALL PLOT(M,J)
      END
```

THE UNMODIFIED DISCRETE ORDINATES ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING PROBLEM PARAMETERS

HALF-RANGE CASE I PROBLEM.
SLAB WIDTH=10.0 CM.
TOTAL MACROSCOPIC CROSS SECTION=1.0000 1/CM.
SCATTERING MACROSCOPIC CROSS SECTION=0.9999 1/CM.
NUMBER OF SPATIAL MESH INCREMENTS= 30 .
SIZE OF EACH SPATIAL MESH INCREMENT=0.33333 CM.
ANGULAR QUADRATURE SET USED=SIX ORDINATE GAUSS.
SOURCE=FIRST COLLISION WITH UNIT STRENGTH.

THE FRACTIONAL DIFFERENCE CONVERGENCE BOUND, 0.10000D-07, WAS MET AFTER 59 ITERATIONS. THIS IMPLIES THAT THE FRACTIONAL ITERATION ERROR IS LESS THAN 0.0001 .

THE INFINITY NORM OF THE ITERATION MATRIX=0.99993 BASED ON THE VALUES OF E(1)/D(1)=0.17754 AND 1/D(1)=0.82246 .

THE CONVERGED ITERATE ANGULAR FLUXES ARE LISTED

X,CM.	MU(1)	MU(2)	MU(3)
0.0	0.0	0.0	0.0
0.3333	0.51073D 00	0.41166D 00	0.32560D 00
0.6667	0.68227D 00	0.66612D 00	0.56021D 00
1.0000	0.72764D 00	0.79689D 00	0.74276D 00
1.3333	0.76418D 00	0.86262D 00	0.84352D 00
1.6667	0.85370D 00	0.89780D 00	0.90514D 00
2.0000	0.87964D 00	0.91869D 00	0.94215D 00
2.3333	0.90358D 00	0.93252D 00	0.96371D 00
2.6667	0.92094D 00	0.94247D 00	0.97560D 00
3.0000	0.94431D 00	0.95001D 00	0.98155D 00
3.3333	0.95409D 00	0.95585D 00	0.98404D 00
3.6667	0.96073D 00	0.96042D 00	0.98452D 00
4.0000	0.96521D 00	0.96398D 00	0.98395D 00
4.3333	0.96835D 00	0.96675D 00	0.98290D 00
4.6667	0.97043D 00	0.96889D 00	0.98171D 00
5.0000	0.97195D 00	0.97052D 00	0.98054D 00
5.3333	0.97295D 00	0.97174D 00	0.97949D 00
5.6667	0.97363D 00	0.97266D 00	0.97859D 00
6.0000	0.97403D 00	0.97333D 00	0.97784D 00
6.3333	0.97436D 00	0.97382D 00	0.97722D 00
6.6667	0.97459D 00	0.97417D 00	0.97672D 00
7.0000	0.97470D 00	0.97441D 00	0.97631D 00
7.3333	0.97477D 00	0.97456D 00	0.97599D 00
7.6667	0.97477D 00	0.97466D 00	0.97572D 00
8.0000	0.97473D 00	0.97472D 00	0.97551D 00
8.3333	0.97477D 00	0.97474D 00	0.97533D 00
8.6667	0.97473D 00	0.97473D 00	0.97518D 00
9.0000	0.97468D 00	0.97471D 00	0.97505D 00
9.3333	0.97463D 00	0.97467D 00	0.97494D 00
9.6667	0.97457D 00	0.97463D 00	0.97483D 00
10.0000	0.97451D 00	0.97458D 00	0.97474D 00

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

APPENDIX D

HALF-RANGE TRANSFORM METHOD PROGRAM AND SAMPLE OUTPUT

```
C  THIS PROGRAM UTILIZES THE EXPONENTIAL TRANSFORM TO ACCELERATE
C  CONVERGENCE OF THE HALF-RANGE CASE I DISCRETE ORDINATES
C  ALGORITHM WITH THE FIRST COLLISION SOURCE.
C  THE VARIABLES EMPLOYED ARE:
C     A=WIDTH OF SLAB IN CM.
C     R=INTEGER GIVING THE NUMBER OF SPATIAL SEGMENTS.
C     M=NUMBER OF SPACE FLUX POINTS, INCLUDING BOTH BOUNDARIES.
C     SIGMA=TOTAL COLLISION CROSS SECTION IN UNITS 1/CM.
C     SIGMAS=MACROSCOPIC SCATTERING CROSS SECTION IN UNITS 1/CM.
C     WEIGHT(J)=WEIGHTS OF THE QUADRATURE SET.
C     MU(J)=DISCRETE SET OF DIRECTION COSINES OF ANGLE WITH
C           RESPECT TO THE X-AXIS.
C     PHI(K,J)=DIRECTIONAL FLUX AT SPATIAL POINTS X(K) DESCRIBING
C              NEUTRONS GOING IN DIRECTION MU(J).
C     MIDSOR(K,J)=TOTAL SOURCE TERM EVALUATED AT SPATIAL MIDPOINTS
C                 AND MU(J).  COMPOSED OF SCATTERING TERM PLUS
C                 THE EXTERNAL SOURCE.
C     MIDPHI(K,J)=DIRECTIONAL FLUX EVALUATED AT SPATIAL MIDPOINTS
C                 DETERMINED FROM PRESENT ITERATION OF PHI(K,J)
C                 AND USED TO CALCULATE NEXT ITERATE SCATTERING
C                 SOURCE TERM.
C     SOURCE(K,J)=EXTERNAL SOURCE AT SPATIAL MIDPOINTS AND MU(J).
C                 THIS TERM DOES NOT CHANGE WITH EACH ITERATION.
C     EPS=A CONVERGENCE CRITERION BOUND FOR FRACTIONAL DIFFERENCES.
C     EPS1=UPPER BOUND ON THE FRACTIONAL ERROR.
C     NORM=INFINITY NORM OF THE ITERATION MATRIX.
C     E(J)=MATRIX ELEMENT OF (D-E) CORRESPONDING TO MU(J).
C     D(J)=MATRIX ELEMENT OF (D-E) CORRESPONDING TO MU(J).
C     ALPHA=ACCELERATION PARAMETER.
C     NOIT=NUMBER OF ITERATIONS TO CONVERGENCE OF UNMODIFIED
C          DISCRETE ORDINATES ALGORITHM.
C
C
      INTEGER R
      REAL*8 MU(3),MIDSOR(30,3),MIDPHI(30,3),NORM
      DOUBLE PRECISION PHI(31,3),SOURCE(30,3),SIGMA,SIGMAS,
     1WEIGHT(3),A,DELTA,SUM,SUM1,SUM2,X(31),OLDPHI(31,3),
     1CONV,D(3),E(3),EPS,EPS1,COMP,COMP1,COMP2,ALPHA
      COMMON /STORE/PHI,X,MU
      NOIT=59
      EPS1=0.0001
      A=10.0
      R=30
      M=R+1
      SIGMA=1.0
      SIGMAS=0.9999
      MU(1) = .23862
      MU(2) = .66121
      MU(3) = .93247
      WEIGHT(1) = .46791
      WEIGHT(2) = .36076
      WEIGHT(3) = .17132
      DELTA=A/R
C  THIS PART OF PROGRAM CALCULATES THE OPTIMUM ALPHA.
      ALPHA=0.2
```

```
C  THIS PART OF THE PROGRAM CALCULATES THE NORM OF THE
C  UNMODIFIED ALGORITHM ITERATION MATRIX FOR LATER COMPARISON.
      E(1)=(2.*MU(1)-DELTA*SIGMA)/(2.*DELTA)
      D(1)=(2.*MU(1)+DELTA*SIGMA)/(2.*DELTA)
      J=P-2
      SUM=DABS(E(1)/D(1))
      SUM1=1.0
      SUM2=0.
      DO 25 I=1,J
      SUM1=SUM1*SUM
   25 SUM2=SUM2+SUM1
      SUM1=SUM1*SUM
      NORM=SIGMAS/2.*(2./D(1)*(1.+SUM2)+1./D(1)*SUM1)
      COMP=NORM
      COMP1=E(1)/D(1)
      COMP2=1./D(1)
C THIS PART OF PROGRAM CALCULATES THE ALGORITHM COEFFICIENTS.
      DO 30 I=1,3
      E(I)=(2.*MU(I)-DELTA*SIGMA-DELTA*ALPHA*MU(I))/(2.*DELTA)
   30 D(I)=(2.*MU(I)+DELTA*SIGMA+DELTA*ALPHA*MU(I))/(2.*DELTA)
C  THIS PART OF THE PROGRAM COMPUTES THE NORM OF THE ITERATION
C  MATRIX.  IT DEPENDS UPON MU(1) BEING THE MINIMUM IN THE SET.
      J=R-2
      SUM=DABS(E(1)/D(1))
      SUM1=1.
      SUM2=0.
      DO 45 I=1,J
      SUM1=SUM1*SUM
   45 SUM2=SUM2+SUM1
      SUM1=SUM1*SUM
      NORM=SIGMAS/2.*(2./D(1)*(1.+SUM2)+1./D(1)*SUM1)
C  THIS PART OF PROGRAM CALCULATES THE ALGORITHM CONVERGENCE
C  CRITERION WHICH GUARANTEES THAT FRACTIONAL ERROR IS LESS
C  THAN EPS1.
      EPS=EPS1/(1.-EPS1)*(1.-NORM)
      DO 50 I=1,M
   50 X(I)=(I-1)*DELTA
C  THE TRANSFORM DOMAIN FIRST COLLISION SOURCE IS CALCULATED.
      DO 80 K=1,R
      DO 80 J=1,3
      SUM=(X(K+1)+X(K))/2.
      SOURCE(K,J)=DEXP(-SUM*(ALPHA+SIGMA/MU(J)))
   80 MIDSOR(K,J)=SOURCE(K,J)
      DO 85 J=1,3
      DO 85 K=1,M
   85 PHI(K,J)=0.
C  THIS PART OF THE PROGRAM STARTS THE INNER ITERATION.  NOTE
C  THAT THE INITIAL FLUX VECTOR GUESS IS THE ZERO VECTOR.
      DO 312 INNER=1,1000
      DO 175 J=1,3
      DO 150 K=2,M
      PHI(K,J)=E(J)/D(J)*PHI(K-1,J)+1./D(J)*MIDSOR(K-1,J)
  150 CONTINUE
  175 CONTINUE
C    THIS LOOP COMPUTES THE NEW SPATIAL MID-POINT FLUXES FROM THE PRESENT
```

```
C     ITERATE FLUXES.
      DO 230 J=1,3
      DO 230 K=1,R
  230 MIDPHI(K,J) = ( PHI(K+1,J) + PHI(K,J))/2.
C  CHECK FOR POINTWISE CONVERGENCE IS CONDUCTED.
C  NO CONVERGENCE CHECK IS MADE THE FIRST TIME THROUGH.
      IF (INNER .EQ. 1) GO TO 280
C  THE NEW CONVERGENCE CRITERION, BASED ON THE MAXIMUM DIFFERENCE
C  AND ITERATION MATRIX NORM, IS USED.
C  A TABLE SEARCH IS CONDUCTED FOR THE MAXIMUM DIFFERENCE.
      SUM=0.
      DO 255 K=1,M
      DO 255 J=1,3
      SUM1=PHI(K,J)-OLDPHI(K,J)
      IF(SUM1 .GT. SUM) SUM=SUM1
  255 CONTINUE
      DO 260 K=2,M
      DO 260 J=1,3
      CONV=SUM/OLDPHI(K,J)
      IF(CONV .GT. EPS) GO TO 280
  260 CONTINUE
      GO TO 600
  280 CONTINUE
C  THIS LOOP CALCULATES THE NEW SOURCE.
      DO 290 K=1,R
      DO 290 J=1,3
      SUM1=0.
      DO 285 I=1,3
  285 SUM1=SUM1+WEIGHT(I)*MIDPHI(K,I)
  290 MIDSOR(K,J)=SIGMAS*SUM1+SOURCE(K,J)
      DO 295 K=1,M
      DO 295 J=1,3
  295 OLDPHI(K,J)=PHI(K,J)
  312 CONTINUE
  600 WRITE(6,610)
  610 FORMAT('1',5X,'THE TRANSFORM METHOD ',
     1'ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING',
     1' PROBLEM PARAMETERS')
      WRITE(6,615)
  615 FORMAT('0',19X,'HALF-RANGE CASE I PROBLEM.')
      WRITE(6,620) A,SIGMA,SIGMAS,R
  620 FORMAT(' ',19X,'SLAB WIDTH=',F4.1,' CM.'/20X,'TOTAL ',
     1'MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,'SCATTERI',
     1'NG MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,
     1'NUMBER OF SPATIAL MESH INCREMENTS=',I3,' .')
      WRITE(6,625) DELTA
  625 FORMAT(' ',19X,'SIZE OF EACH SPATIAL MESH INCREMENT=',
     1F7.5,' CM.'/20X,'ANGULAR QUADRATURE SET USED=SIX ',
     1'ORDINATE GAUSS.'/20X,'SOURCE=FIRST COLLISION',
     1' WITH UNIT STRENGTH.')
      WRITE(6,626) COMP1,COMP2,COMP
  626 FORMAT(' ',19X,'UNMODIFIED ALGORITHM COEFFICIENT E(1)/D(',
     1'1)=',F8.5,' .'/20X,'UNMODIFIED ALGORITHM COEFFICIENT 1/',
     1'D(1)=',F7.5,' .'/20X,'UNMODIFIED ALGORITHM ITERATION MAT',
     1'RIX NORM=',F7.5,' .')
```

```
      WRITE(6,627) NOIT,ALPHA
  627 FORMAT(' ',19X,'NUMBER OF ITERATIONS TO CONVERGENCE OF',
     1' UNMODIFIED ALGORITHM=',I4,' .'/20X,'ACCELERATION ',
     1'PARAMETER=',F7.5,' .')
      WRITE(6,630) EPS,INNER,EPS1
  630 FORMAT('0',4X,'THE FRACTIONAL DIFFERENCE CONVERGENCE',
     1' BOUND,',E12.5,', WAS MET AFTER ',I5,' ITERATIONS.',
     1' THIS IMPLIES '/5X,'THAT THE FRACTIONAL ITERATION',
     1' ERROR IS LESS THAN ',F6.4,' .')
      SUM=E(1)/D(1)
      SUM1=1./D(1)
      WRITE(6,640) NORM,ALPHA,SUM,SUM1
  640 FORMAT('0',4X,'THE INFINITY NORM OF THE TRANSFORM',
     1' DOMAIN ITERATION MATRIX=',F7.5,' BASED ON THE VALUES',
     1' OF ALPHA=',F7.4,','/5X,'E(1)/D(1)=',F7.5,' AND 1/D(1)=',
     1F7.5,' .')
C  TRANSFORM TO THE SOLUTION DOMAIN IS CONDUCTED.
      DO 645 J=1,3
      DO 645 K=1,M
  645 PHI(K,J)=PHI(K,J)*DEXP(ALPHA*X(K))
      WRITE(6,650)
  650 FORMAT('0',30X,'THE CONVERGED ITERATE SOLUTION ANGULAR',
     1' FLUXES ARE LISTED')
      WRITE(6,660)
  660 FORMAT(' ',32X,'X,CM.',11X,'MU(1)',15X,'MU(2)',
     115X,'MU(3)')
      DO 665 K=1,M
  665 WRITE(6,670) X(K),(PHI(K,J),J=1,3)
  670 FORMAT(' ',30X,F7.4,3(8X,E12.5))
      WRITE(6,680)
  680 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
      DO 685 J=1,3
  685 CALL PLOT(M,J)
      WRITE(6,690)
  690 FORMAT('1',31X,'THE TRANSFORM DOMAIN PREVIOUS ITERATE ANG',
     1'ULAR FLUXES ARE LISTED')
      WRITE(6,700)
  700 FORMAT(' ',32X,'X,CM.',11X,'MU(1)',15X,'MU(2)',
     115X,'MU(3)')
      DO 705 K=1,M
  705 WRITE(6,710) X(K),(OLDPHI(K,J),J=1,3)
  710 FORMAT(' ',30X,F7.4,3(8X,E12.5))
      WRITE(6,720)
  720 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
      DO 900 J=1,3
      DO 900 K=1,M
  900 PHI(K,J)=OLDPHI(K,J)
      DO 910 J=1,3
  910 CALL PLOT(M,J)
      END
```

THE TRANSFORM METHOD ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING PROBLEM PARAMETERS

HALF-RANGE CASE I PROBLEM.
SLAB WIDTH=10.0 CM.
TOTAL MACROSCOPIC CROSS SECTION=1.0000 1/CM.
SCATTERING MACROSCOPIC CROSS SECTION=0.9999 1/CM.
NUMBER OF SPATIAL MESH INCREMENTS= 30 .
SIZE OF EACH SPATIAL MESH INCREMENT=0.33333 CM.
ANGULAR QUADRATURE SET USED=SIX ORDINATE GAUSS.
SOURCE=FIRST COLLISION WITH UNIT STRENGTH.
UNMODIFIED ALGORITHM COEFFICIENT E(1)/D(1)= 0.17754 .
UNMODIFIED ALGORITHM COEFFICIENT 1/D(1)=0.82246 .
UNMODIFIED ALGORITHM ITERATION MATRIX NORM=0.99990 .
NUMBER OF ITERATIONS TO CONVERGENCE OF UNMODIFIED ALGORITHM= 59 .
ACCELERATION PARAMETER=0.20000 .

THE FRACTIONAL DIFFERENCE CONVERGENCE BOUND, 0.456530-05, WAS MET AFTER 48 ITERATIONS. THIS IMPLIES THAT THE FRACTIONAL ITERATION ERROR IS LESS THAN 0.0001 .

THE INFINITY NORM OF THE TRANSFORM DOMAIN ITERATION MATRIX=0.95435 BASED ON THE VALUES OF ALPHA= 0.2000. E(1)/D(1)=0.15487 AND 1/D(1)=0.80663 .

THE CONVERGED ITERATE SOLUTION ANGULAR FLUXES ARE LISTED

X,CM.	MU(1)	MU(2)	MU(3)
0.0	0.0	0.0	0.0
0.3333	0.612920 00	0.411490 00	0.327460 00
0.6667	0.681660 00	0.667100 00	0.580540 00
1.0000	0.726920 00	0.797650 00	0.743410 00
1.3333	0.784030 00	0.863110 00	0.844210 00
1.6667	0.837300 00	0.898110 00	0.905760 00
2.0000	0.879140 00	0.918940 00	0.942740 00
2.3333	0.909440 00	0.932770 00	0.964220 00
2.6667	0.930450 00	0.942740 00	0.976050 00
3.0000	0.944790 00	0.950300 00	0.981970 00
3.3333	0.954520 00	0.956150 00	0.984360 00
3.6667	0.961120 00	0.960720 00	0.984520 00
4.0000	0.965610 00	0.964290 00	0.984210 00
4.3333	0.968670 00	0.967040 00	0.983140 00
4.6667	0.970760 00	0.969160 00	0.981920 00
5.0000	0.972190 00	0.970760 00	0.980730 00
5.3333	0.973150 00	0.971970 00	0.979660 00
5.6667	0.973820 00	0.972860 00	0.978740 00
6.0000	0.974250 00	0.973500 00	0.977970 00
6.3333	0.974530 00	0.973970 00	0.977330 00
6.6667	0.974730 00	0.974290 00	0.976810 00
7.0000	0.974790 00	0.974500 00	0.976360 00
7.3333	0.974830 00	0.974630 00	0.976040 00
7.6667	0.974830 00	0.974700 00	0.975750 00
8.0000	0.974800 00	0.974730 00	0.975510 00
8.3333	0.974760 00	0.974730 00	0.975310 00
8.6667	0.974690 00	0.974700 00	0.975140 00
9.0000	0.974620 00	0.974650 00	0.974990 00
9.3333	0.974550 00	0.974590 00	0.974850 00
9.6667	0.974440 00	0.974520 00	0.974720 00
10.0000	0.974330 00	0.974440 00	0.974600 00

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

THE TRANSFORM DOMAIN PREVIOUS ITERATE ANGULAR FLUXES ARE LISTED

X,CM.	MU(1)	MU(2)	MU(3)
0.0	0.0	0.0	0.0
0.3333	0.57339D 00	0.38495D 00	0.30635D 00
0.6667	0.59657D 00	0.58383D 00	0.50608D 00
1.0000	0.59515D 00	0.65306D 00	0.60865D 00
1.3333	0.60051D 00	0.65108D 00	0.64661D 00
1.6667	0.59495D 00	0.64353D 00	0.64902D 00
2.0000	0.58933D 00	0.61599D 00	0.63193D 00
2.3333	0.57030D 00	0.58493D 00	0.60465D 00
2.6667	0.54585D 00	0.55306D 00	0.57260D 00
3.0000	0.51851D 00	0.52155D 00	0.53892D 00
3.3333	0.49007D 00	0.49091D 00	0.50540D 00
3.6667	0.46163D 00	0.46144D 00	0.47301D 00
4.0000	0.43328D 00	0.43328D 00	0.44223D 00
4.3333	0.40718D 00	0.40650D 00	0.41326D 00
4.6667	0.38174D 00	0.38111D 00	0.38613D 00
5.0000	0.35765D 00	0.35712D 00	0.36079D 00
5.3333	0.33492D 00	0.33451D 00	0.33715D 00
5.6667	0.31353D 00	0.31322D 00	0.31511D 00
6.0000	0.29344D 00	0.29321D 00	0.29456D 00
6.3333	0.27459D 00	0.27443D 00	0.27538D 00
6.6667	0.25693D 00	0.25682D 00	0.25746D 00
7.0000	0.24038D 00	0.24031D 00	0.24077D 00
7.3333	0.22489D 00	0.22484D 00	0.22516D 00
7.6667	0.21038D 00	0.21036D 00	0.21058D 00
8.0000	0.19681D 00	0.19680D 00	0.19695D 00
8.3333	0.18411D 00	0.18410D 00	0.18421D 00
8.6667	0.17222D 00	0.17222D 00	0.17230D 00
9.0000	0.16110D 00	0.16111D 00	0.16116D 00
9.3333	0.15070D 00	0.15072D 00	0.15075D 00
9.6667	0.14097D 00	0.14098D 00	0.14101D 00
10.0000	0.13187D 00	0.13188D 00	0.13190D 00

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

```
THE TRANSFORM METHOD ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING PROBLEM PARAMETERS

        HALF-RANGE CASE I PROBLEM.
        SLAB WIDTH=10.0 CM.
        TOTAL MACROSCOPIC CROSS SECTION=1.0000 1/CM.
        SCATTERING MACROSCOPIC CROSS SECTION=0.9999 1/CM.
        NUMBER OF SPATIAL MESH INCREMENTS= 30 .
        SIZE OF EACH SPATIAL MESH INCREMENT=0.33333 CM.
        ANGULAR QUADRATURE SET USED=SIX ORDINATE GAUSS.
        SOURCE=FIRST COLLISION WITH UNIT STRENGTH.
        UNMODIFIED ALGORITHM COEFFICIENT E(1)/D(1)= 0.17754 .
        UNMODIFIED ALGORITHM COEFFICIENT 1/D(1)=0.82246 .
        UNMODIFIED ALGORITHM ITERATION MATRIX NORM=0.99990 .
        NUMBER OF ITERATIONS TO CONVERGENCE OF UNMODIFIED ALGORITHM=  59 .
        ACCELERATION PARAMETER=1.80924 .

THE FRACTIONAL DIFFERENCE CONVERGENCE BOUND, 0.301640D-04, WAS MET AFTER    45 ITERATIONS. THIS IMPLIES
THAT THE FRACTIONAL ITERATION ERROR IS LESS THAN 0.0001 .

THE INFINITY NORM OF THE TRANSFORM DOMAIN ITERATION MATRIX=0.69839 BASED ON THE VALUES OF ALPHA= 1.8092,
E(1)/D(1)=0.00000 AND 1/D(1)=0.69346 .

              THE CONVERGED ITERATE SOLUTION ANGULAR FLUXES ARE LISTED
```

X,CM.	MU(1)	MU(2)	MU(3)
0.0	0.0	0.0	0.0
0.3333	0.649220 00	0.424380 00	0.335140 00
0.6667	0.67[illegible]D 00	0.689830 00	0.600300 00
1.0000	0.720560 00	0.8[illegible]D 00	0.758060 00
1.3333	0.776720 00	0.857060 00	0.845940 00
1.6667	0.821520 00	0.876330 00	0.891630 00
2.0000	0.849440 00	0.882080 00	0.911440 00
2.3333	0.863340 00	0.880830 00	0.915320 00
2.6667	0.867120 00	0.875420 00	0.909530 00
3.0000	0.864150 00	0.867240 00	0.898080 00
3.3333	0.856770 00	0.857130 00	0.883480 00
3.6667	0.846560 00	0.845630 00	0.867250 00
4.0000	0.834540 00	0.833140 00	0.850450 00
4.3333	0.821470 00	0.819950 00	0.833490 00
4.6667	0.8076[illegible]D 00	0.806300 00	0.816720 00
5.0000	0.793490 00	0.792370 00	0.800260 00
5.3333	0.779140 00	0.778290 00	0.784240 00
5.6667	0.764870 00	0.764180 00	0.768610 00
6.0000	0.750670 00	0.750110 00	0.753380 00
6.3333	0.736520 00	0.736140 00	0.738550 00
6.6667	0.722560 00	0.722310 00	0.724070 00
7.0000	0.708840 00	0.708650 00	0.709940 00
7.3333	0.695310 00	0.695190 00	0.696130 00
7.6667	0.682010 00	0.681950 00	0.682630 00
8.0000	0.668950 00	0.668920 00	0.669420 00
8.3333	0.656120 00	0.656110 00	0.656480 00
8.6667	0.643520 00	0.643530 00	0.643810 00
9.0000	0.631160 00	0.631190 00	0.631390 00
9.3333	0.619030 00	0.619070 00	0.619220 00
9.6667	0.607130 00	0.607170 00	0.607290 00
10.0000	0.595450 00	0.595500 00	0.595590 00

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

THE TRANSFORM DOMAIN PREVIOUS ITERATE ANGULAR FLUXES ARE LISTED

X,CM.	MU(1)	MU(2)	MU(3)
0.0	0.0	0.0	0.0
0.3333	0.35[illegible]21D 00	0.23219D 00	0.18336D 00
0.6667	0.20296D 00	0.20650D 00	0.17470D 00
1.0000	0.11501D 00	0.13235D 00	0.12414D 00
1.3333	0.5[illegible]D-01	0.75799D-01	0.75603D-01
1.6667	0.4[illegible]76D-01	0.42963D-01	0.45714D-01
2.0000	0.227[illegible]5D-01	0.23661D-01	0.24443D-01
2.3333	0.125[illegible]D-01	0.12927D-01	0.13433D-01
2.6667	0.69526D-02	0.70292D-02	0.73031D-02
3.0000	0.37964D-02	0.38099D-02	0.39454D-02
3.3333	0.20593D-02	0.20602D-02	0.21235D-02
3.6667	0.11133D-02	0.11121D-02	0.11405D-02
4.0000	0.6[illegible]45D-03	0.59945D-03	0.61190D-03
4.3333	0.32336D-03	0.32278D-03	0.32811D-03
4.6667	0.17195D-03	0.17356D-03	0.17591D-03
5.0000	0.93[illegible]55D-04	0.93373D-04	0.94305D-04
5.3333	0.50237D-04	0.50179D-04	0.50562D-04
5.6667	0.26781D-04	0.26956D-04	0.27113D-04
6.0000	0.14487D-04	0.14477D-04	0.14540D-04
6.3333	0.77771D-05	0.77732D-05	0.77986D-05
6.6667	0.41745D-05	0.41730D-05	0.41632D-05
7.0000	0.22406D-05	0.22400D-05	0.22441D-05
7.3333	0.12025D-05	0.12023D-05	0.12034D-05
7.6667	0.54532D-06	0.64526D-06	0.64591D-06
8.0000	0.34631D-06	0.34629D-06	0.34655D-06
8.3333	0.18564D-06	0.18564D-06	0.18594D-06
8.6667	0.99725D-07	0.99727D-07	0.99770D-07
9.0000	0.53514D-07	0.53516D-07	0.53534D-07
9.3333	0.28716D-07	0.28718D-07	0.28725D-07
9.6667	0.15409D-07	0.15410D-07	0.15413D-07
10.0000	0.82684D-08	0.82691D-08	0.82705D-08

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

APPENDIX E

DISCRETIZATION ERROR IMPROVEMENT PROGRAM AND SAMPLE OUTPUT

```
C  THIS PROGRAM USES EXPONENTIAL TRANSFORM IN AN ATTEMPT TO IMPROVE
C  THE DISCRETIZATION ERROR.
C  THE VARIABLES EMPLOYED ARE:
C    A=WIDTH OF SLAB IN CM.
C    R=INTEGER GIVING THE NUMBER OF SPATIAL SEGMENTS.
C    N=NUMBER OF SPACE FLUX POINTS, INCLUDING BOTH BOUNDARIES.
C    SIGMA=TOTAL COLLISION CROSS SECTION IN UNITS 1/CM.
C    SIGMAS=MACROSCOPIC SCATTERING CROSS SECTION IN UNITS 1/CM.
C    WEIGHT(J)=WEIGHTS OF THE QUADRATURE SET.
C    MU(J)=DISCRETE SET OF DIRECTION COSINES OF ANGLE WITH
C          RESPECT TO THE X-AXIS.
C    PHI(K,J)=DIRECTIONAL FLUX AT SPATIAL POINTS X(K) DESCRIBING
C             NEUTRONS GOING IN DIRECTION MU(J).
C    MIDSOR(K,J)=TOTAL SOURCE TERM EVALUATED AT SPATIAL MIDPOINTS
C                AND MU(J).  COMPOSED OF SCATTERING TERM PLUS
C                THE EXTERNAL SOURCE.
C    MIDPHI(K,J)=DIRECTIONAL FLUX EVALUATED AT SPATIAL MIDPOINTS
C                DETERMINED FROM PRESENT ITERATION OF PHI(K,J)
C                AND USED TO CALCULATE NEXT ITERATE SCATTERING
C                SOURCE TERM.
C    SOURCE(K,J)=EXTERNAL SOURCE AT SPATIAL MIDPOINTS AND MU(J).
C                THIS TERM DOES NOT CHANGE WITH EACH ITERATION.
C    EPS=CONVERGENCE CRITERION BOUND FOR FRACTIONAL DIFFERENCES.
C    ALPHA=CONVERGENCE ACCELERATION PARAMETER.
C    X(I)=SLAB SPATIAL MESH POINT.
C    F(J)=COMPONENT OF ALGORITHM COEFFICIENT.
C    D(J)=COMPONENT OF ALGORITHM COEFFICIENT.
C    PP1(K,J)=STORAGE FOR R=30 UNMODIFIED ALGORITHM SOLUTION.
C    PP2(K,J)=STORAGE FOR R=30 TRANSFORM METHOD SOLUTION.
C    PP3(K,J)=STORAGE FOR R=600 UNMODIFIED ALGORITHM SOLUTION.
C    DIF1(K,J)=DIFFERENCE BETWEEN UNMODIFIED R=600 RESULT
C              AND UNMODIFIED R=30 RESULT.
C    DIF2(K,J)=DIFFERENCE BETWEEN R=30 TRANSFORM METHOD RESULT
C              AND R=30 UNMODIFIED RESULT.
C    DIF3(K,J)=DIFFERENCE BETWEEN R=600 UNMODIFIED RESULT
C              AND R=30 TRANSFORM METHOD RESULT.
C    MP3(K,J)=MIDPHI(K,J) FOR R=600 ALGORITHM.
C    MS(K,J)=MIDSOR(K,J) FOR R=600 ALGORITHM.
C    SS(K,J)=SOURCE(K,J) FOR R=600 ALGORITHM.
C    OPP3(K,J)=OLDPHI(K,J) FOR R=600 ALGORITHM.
C
C
      INTEGER R
      REAL*8 MU(6),MIDSOR(30,6),MIDPHI(30,6)
      DOUBLE PRECISION PHI(31,6),SOURCE(30,6),SIGMA,SIGMAS,
     1WEIGHT(6),A,DELTA,SUM,SUM1,X(601),OLDPHI(31,6),CONV
      DOUBLE PRECISION ALPHA,F(6),D(6),PP2(31,6),PP1(31,6),
     1PP3(601,6),MP3(600,6),MS(600,6),SS(600,6),DIF1(31,6),
     1DIF2(31,6),DIF3(31,6),OPP3(601,6),BULK1(6),BULK2(6)
      NUMB=1
      EPS=0.0001
      A=10.
   10 CONTINUE
      ALPHA=0.1
      IF(NUMB .LT. 3) R=30
```

```
      IF(NUMB .EQ. 3) P=600
      M=P+1
      SIGMA=1.0
      SIGMAS=0.2
      MU(1) = .23862
      MU(2) = .66121
      MU(3) = .93247
      MU(4) = -.23862
      MU(5) = -.66121
      MU(6) = -.93247
      WEIGHT(1) = .46791
      WEIGHT(2) = .36076
      WEIGHT(3) = .17132
      WEIGHT(4) = .46791
      WEIGHT(5) = .36076
      WEIGHT(6) = .17132
      DELTA=1/P
      DO 50 I=1,M
   50 Y(I)=(I-1)*DELTA
      IF(NUMB .EQ. 1) ALPHA=0.
      IF(NUMB .EQ. 3) ALPHA=0.
C  THIS PART OF PROGRAM CALCULATES THE ALGORITHM COEFFICIENTS.
      DO 20 J=1,3
      E(J)=(2.*MU(J)-DELTA*SIGMA+DELTA*ALPHA*MU(J))/(2.*DELTA)
   20 D(J)=(2.*MU(J)+DELTA*SIGMA-DELTA*ALPHA*MU(J))/(2.*DELTA)
      DO 30 J=4,6
      SUM=DABS(MU(J))
      E(J)=(2.*SUM-DELTA*SIGMA-DELTA*ALPHA*SUM)/(2.*DELTA)
   30 D(J)=(2.*SUM+DELTA*SIGMA+DELTA*ALPHA*SUM)/(2.*DELTA)
      IF(NUMB .EQ. 3) GO TO 760
C  THERE IS NO SOURCE TRANSFORMATION FOR THIS PROBLEM.
      DO 80 J=1,6
      DO 80 K=1,M
      SOURCE(K,J)=0.
   80 MIDSOR(K,J)=0.
      DO 85 J=1,6
      DO 85 K=1,M
   85 PHI(K,J)=0.
C  THIS PART OF THE PROGRAM STARTS THE INNER ITERATION.  NOTE
C  THAT THE INITIAL FLUX VECTOR GUESS IS THE ZERO VECTOR.
C  THE BOUNDARY CONDITIONS ARE SET.
      DO 90 J=1,3
   90 PHI(1,J)=1.0
      DO 312 INNER=1,200
      DO 175 J=1,3
      DO 150 K=2,M
      PHI(K,J)=E(J)/D(J)*PHI(K-1,J)+1./D(J)*MIDSOR(K-1,J)
  150 CONTINUE
  175 CONTINUE
      DO 190 J=4,6
      DO 185 K=1,P
      N = M-K
      PHI(N,J)=E(J)/D(J)*PHI(N+1,J)+1./D(J)*MIDSOR(N,J)
  185 CONTINUE
  190 CONTINUE
```

```
C     THIS LOOP COMPUTES THE NEW SPATIAL MID-POINT FLUXES FROM THE PRESENT
C     ITERATE FLUXES.
          DO 230 J=1,6
          DO 230 K=1,8
  230   MIDPHI(K,J) = ( PHI(K+1,J) + PHI(K,J))/2.
C  CHECK FOR POINTWISE CONVERGENCE IS CONDUCTED.
C  NO CONVERGENCE CHECK IS MADE THE FIRST TIME THROUGH.
          IF (INNER .EQ. 1) GO TO 280
          DO 260 K=2,M
          DO 260 J=1,3
          CONV=DABS(1.-PHI(K,J)/OLDPHI(K,J))
          IF(CONV .GT. EPS) GO TO 280
  260   CONTINUE
          DO 270 K=1,8
          DO 270 J=4,6
          CONV=DABS(1.-PHI)K,J)/OLDPHI(K,J))
          IF(CONV .GT. EPS) GO TO 280
  270   CONTINUE
          GO TO 600
  280   CONTINUE
C  THIS LOOP CALCULATES THE NEW SOURCE.
          DO 290 K=1,8
          DO 290 J=1,6
          SUM1=0.
          DO 285 I=1,6
  285   SUM1=SUM1+WEIGHT(I)*MIDPHI(K,I)
  290   MIDSOR(K,J)=SIGMAS/2. * SUM1 + SOURCE(K,J)
          DO 295 K=1,M
          DO 295 J=1,6
  295   OLDPHI(K,J)=PHI(K,J)
  312   CONTINUE
  600   IF(NUMB .GT. 1) GO TO 602
          DO 601 J=1,6
          DO 601 K=1,M
  601   PPI(K,J)=PHI)K,J)
          GO TO 750
  602   CONTINUE
          DO 603 J=1,6
          DO 603 K=1,M
  603   PPI)K,J)=PHI)K,J)*DEXP(-ALPHA*X)K))
          WRITE(6,610)
  610   FORMAT('1',5X,'THE TRANSFORM METHOD ',
       1'ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING',
       1' PROBLEM PARAMETERS')
          WRITE(6,620) A,SIGMA,SIGMAS,M
  620   FORMAT('0',19X,'SLABWIDTH=',F6.1,' CM.'/20X,'TOTAL ',
       1'MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,'SCATTERI',
       1'NG MACROSCOPIC CROSS SECTION=',F6.4,' 1/CM.'/20X,
       1'NUMBER OF SPATIAL MESH INCREMENTS=',I3,' .')
          WRITE(6,625) DELTA
  625   FORMAT(' ',19X,'SIZE OF EACH SPATIAL MESH INCREMENT=',
       1F7.5,' CM.'/20X,'ANGULAR QUADRATURE SET USED=SIX ',
       1'POINT GAUSS-LEGENDRE.'/20X,'SOURCE=ZERO.')
          WRITE(6,626) ALPHA
  626   FORMAT(' ',19X,'ACCELERATION PARAMETER ALPHA=',F6.3,' .')
```

```
      WRITE(6,630) EPS,INNER
  630 FORMAT('0',4X,'THE FRACTIONAL DIFFERENCE CONVERGENCE',
     1' BOUND,',E12.5,', WAS MET AFTER ',I5,' ITERATIONS.')
C  THE SOLUTION IS TRANSFORMED.
      DO 640 K=1,M
      DO 640 J=1,6
  640 PHI(K,J)=PHI(K,J)*DEXP(-ALPHA*X(K))
      WRITE(6,650)
  650 FORMAT('0',30X,'THE CONVERGED ITERATE ANGULAR FLUXES ',
     1'ARE LISTED')
      WRITE(6,660)
  660 FORMAT(' ',10X,'X,CM.',7X,'MU(1)',9X,'MU(2)',9X,'MU(3)',
     19X,'MU(4)',9X,'MU(5)',9X,'MU(6)')
      DO 665 K=1,M
  665 WRITE(6,670) X(K),(PHI(K,J),J=1,6)
  670 FORMAT(' ',8X,F7.4,6(2X,F12.5))
      WRITE(6,680)
  680 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
      WRITE(6,690)
  690 FORMAT('1',10X,'THE TRANSFORM DOMAIN ANGULAR FLUXES ',
     1'ARE LISTED')
      WRITE(6,700)
  700 FORMAT(' ',10X,'X,CM.',7X,'MU(1)',9X,'MU(2)',9X,'MU(3)',
     19X,'MU(4)',9X,'MU(5)',9X,'MU(6)')
      DO 705 K=1,M
  705 WRITE(6,710) X(K),(OLDPHI(K,J),J=1,6)
  710 FORMAT(' ',8X,F7.4,6(2X,F12.5))
      WRITE(6,720)
  720 FORMAT('0',5X,'NOTE THAT D 01 MEANS TEN TO THE POWER',
     1' ONE.')
  750 NUMB=NUMB+1
      GO TO 10
  760 CONTINUE
      DO 780 J=1,6
      DO 780 K=1,R
      SS(K,J)=0.
  780 MS(K,J)=0.
      DO 785 J=1,6
      DO 785 K=1,M
  785 PP3(K,J)=0.
C  THIS PART OF PROGRAM STARTS THE INNER ITERATION. NOTE
C  THAT THE INITIAL FLUX VECTOR GUESS IS THE ZERO VECTOR.
C  THE LEFT BOUNDARY CONDITION IS SET.
      DO 790 J=1,3
  790 PP3(1,J)=1.0
      DO 900 INNER=1,200
      DO 810 J=1,3
      DO 805 K=2,M
      PP3(K,J)=F(J)/D(J)*PP3(K-1,J)+1./D(J)*MS(K-1,J)
  805 CONTINUE
  810 CONTINUE
      DO 920 J=4,6
      DO 815 K=1,R
      N=M-K
```

```
          PP3(N,J)=F(J)/D(J)*PP3(N+1,J)+1./D(J)*MS(N,J)
  815     CONTINUE
  820     CONTINUE
C   THIS LOOP COMPUTES THE NEW SPATIAL MIDPOINT FLUXES FROM THE
C   PRESENT ITERATE FLUXES.
          DO 830 J=1,6
          DO 830 K=1,R
  830     MP3(K,J)=(PP3(K+1,J)+PP3(K,J))/2.
C   CHECK FOR POINTWISE CONVERGENCE IS CONDUCTED.
C   NO CONVERGENCE CHECK IS MADE THE FIRST TIME THROUGH.
          IF(INNER .EQ. 1) GO TO 880
          DO 860 K=2,M
          DO 860 J=1,3
          CONV=DABS(1.-PP3(K,J)/OPP3(K,J))
          IF(CONV .GT. EPS) GO TO 880
  860     CONTINUE
          DO 870 K=1,R
          DO 870 J=4,6
          CONV=DABS(1.-PP3(K,J)/OPP3(K,J))
          IF(CONV .GT. EPS) GO TO 880
  870     CONTINUE
          GO TO 900
  880     CONTINUE
C   THIS LOOP CALCULATES THE NEW SOURCE.
          DO 890 K=1,R
          DO 890 J=1,6
          SUM1=0.
          DO 885 I=1,6
  885     SUM1=SUM1+WEIGHT(I)*MP3(K,I)
  890     MS(K,J)=SIGMAS/2.*SUM1+SS(K,J)
          DO 895 K=1,M
          DO 895 J=1,6
  895     OPP3(K,J)=PP3(K,J)
  800     CONTINUE
  900     WRITE(6,905)
  905     FORMAT('1',30X,'THE UNMODIFIED DISCRETE ORDINATES ',
         1'RESULT FOR R=30 ARE LISTED')
          DO 907 I=1,31
  907     X(I)=(I-1)*1./3.
          WRITE(6,910)
  910     FORMAT(' ',10X,'X,CM.',7X,'MU(1)',9X,'MU(2)',9X,'MU(3)',
         19X,'MU(4)',9X,'MU(5)',9X,'MU(6)')
          DO 915 K=1,31
  915     WRITE(6,917) X(K),(PP1(K,J),J=1,6)
  917     FORMAT(' ',9X,F7.4,6(2X,F12.5))
          WRITE(6,920)
  920     FORMAT('1',30X,'THE UNMODIFIED DISCRETE ORDINATES  ',
         1'RESULT FOR R=600 ARE LISTED')
          DO 922 I=1,601
  922     X(I)=(I-1)*10./600.
          WRITE(6,925)
  925     FORMAT(' ',10X,'X,CM.',7X,'MU(1)',9X,'MU(2)',9X,'MU(3)',
         19X,'MU(4)',9X,'MU(5)',9X,'MU(6)')
          DO 930 K=1,M,20
  930     WRITE(6,931) X(K),(PP3(K,J),J=1,6)
```

```
  931 FORMAT(' ',8X,F7.4,6(2X,F12.5))
      DO 935 I=1,31
  935 X(I)=(I-1)*10./30.
C  THIS PART OF PROGRAM COMPUTES DIFFERENCES FOR COMPARISON
C  OF RESULTS.
      SUM=0.
      SUM1=0.
      DO 942 J=1,6
      BULK1(J)=0.
      BULK2(J)=0.
      DO 941 K=1,31
      DIF1(K,J)=PP3(((K-1)*20+1),J)-PP1(K,J)
      BULK1(J)=BULK1(J)+DABS(DIF1(K,J))
      DIF2(K,J)=PP2(K,J)-PP1(K,J)
      DIF3(K,J)=PP3(((K-1)*20+1),J)-PP2(K,J)
  941 BULK2(J)=BULK2(J)+DABS(DIF3(K,J))
      SUM=SUM+BULK1(J)
  942 SUM1=SUM1+BULK2(J)
      DO 970 J=1,6
      WRITE(6,943) J
  943 FORMAT('1',10X,'COMPARISON OF RESULTS FOR THREE ',
     1'DIFFERENT ALGORITHMS ARE MADE FOR MU(',
     1I1,') PART OF SOLUTION')
      WRITE(6,945)
  945 FORMAT('0',2X,'X,CM.',10X,'UNMODIFIED',5X,'TRANSFORM',
     1' METHOD',3X,'UNMODIFIED',7X,'UNMODIFIED',5X,'TRANSFORM',
     1' METHOD',3X,'TRANSFORM METHOD')
      WRITE(6,948)
  948 FORMAT(' ',16X,'R=30 SOLUTION',5X,'R=30 SOLUTION',2X,
     1'R=600 SOLUTION',5X,'DIFFERENCE',3X,'DIFFERENCE',7X,
     1'DIFFERENCE')
      WRITE(6,951) (X(K),PP1(K,J),PP2(K,J),PP3(((K-1)*20+1),J),
     1DIF1(K,J),DIF2(K,J),DIF3(K,J),K=1,31)
  951 FORMAT((F8.4,2X,6(F12.5,5X)))
      WRITE(6,952)
  952 FORMAT(' ',2X,'UNMODIFIED DIFFERENCE=(UNMODIFIED R=600',
     1' SOLUTION)-(UNMODIFIED R=30 SOLUTION).'/2X,'OBSERVE',
     1' THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION ',
     1'VALUES.')
      WRITE(6,953)
  953 FORMAT(' ',2X,'TRANSFORM METHOD DIFFERENCE IN NEXT TO ',
     1'LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODI',
     1'FIED R=30 SOLUTION).'/2X,'A POSITIVE QUANTITY HERE ',
     1'MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.')
      WRITE(6,954)
  954 FORMAT(' ',2X,'TRANSFORM METHOD DIFFERENCE IN LAST COL',
     1'UMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30',
     1' SOLUTION).')
      WRITE(6,955)
  955 FORMAT('0',2X,'THE KEY TO INTERPRETING THE TABLE IS:',
     1/2X,'1.  THE FINER SPATIAL MESH, R=600, PRODUCES LARGER',
     1' SOLUTION VALUES THAN THE R=30 MESH.')
      WRITE(6,956)
  956 FORMAT(' ',1X,'2.  A POSITIVE QUANTITY IN COLUMNS 7 AND',
     1' 8 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZA',
```

```
     1'TION ERROR.'/2X,'3.  A NEGATIVE QUANTITY IN COLUMN 7 M',
     1'EANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.')
      WRITE(6,957)
957   FORMAT(' ',1X,'4.  A POSITIVE QUANTITY IN COLUMN 7 AND',
     1' NEGATIVE IN COLUMN 8 MEANS THE TRANSFORM PRODUCED A SO',
     1'LUTION IN THE DIRECTION'/6X,'OF IMPROVED DISCRETIZATION',
     1' ERROR BUT OVERSHOT THE R=600 SOLUTION.')
      WRITE(6,958) J,BULK1(J)
958   FORMAT('0',1X,'THE BULK SUM OF ABSOLUTE DIFFERENCES ',
     1'IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(',I1,
     1') IS',E12.5,' .')
      WRITE(6,959) J,BULK2(J)
959   FORMAT(' ',1X,'THE BULK SUM OF ABSOLUTE DIFFERENCES',
     1' IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR',
     1' MU(',I1,') IS',E12.5,' .')
      WRITE(6,960) SUM,SUM1
960   FORMAT('0',1X,'THE TOTAL BULK SUM OF ABSOLUTE DIFFEREN',
     1'CES IN R=600 SOLUTION AND R=30 SOLUTION IS ',E12.5,
     1' .'/2X,'THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN',
     1' R=600 AND TRANSFORM METHOD SOLUTIONS IS',E12.5,' .')
970   CONTINUE
      END
```

THE TRANSFORM METHOD ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING PROBLEM PARAMETERS

SLABWIDTH=10.0 CM.
TOTAL MACROSCOPIC CROSS SECTION=1.0000 1/CM.
SCATTERING MACROSCOPIC CROSS SECTION=0.2000 1/CM.
NUMBER OF SPATIAL MESH INCREMENTS= 30 .
SIZE OF EACH SPATIAL MESH INCREMENT=0.33333 CM.
ANGULAR QUADRATURE SET USED=SIX ORDINATE GAUSS-LEGENDRE.
SOURCE=ZERO.
ACCELERATION PARAMETER ALPHA= 0.500 .

THE FRACTIONAL DIFFERENCE CONVERGENCE BOUND, 0.10000E-03, WAS MET AFTER 10 ITERATIONS.

THE CONVERGED ITERATE ANGULAR FLUXES ARE LISTED

X.CM.	MU(1)	MU(2)	MU(3)	MU(4)	MU(5)	MU(6)
0.0	0.10000D 01	0.10000D 01	0.10000D 01	0.69962D-01	0.45799D-01	0.37758D-01
0.3333	0.24215D 00	0.63221D 00	0.72184D 00	0.36237D-01	0.24492D-01	0.20813D-01
0.6667	0.83793D-01	0.39605D 00	0.51625D 00	0.21644D-01	0.15223D-01	0.12816D-01
1.0000	0.35251D-01	0.24760D 00	0.36780D 00	0.13785D-01	0.97998D-02	0.82759D-02
1.3333	0.15765D-01	0.15485D 00	0.26150D 00	0.90199D-02	0.64465D-02	0.54535D-02
1.6667	0.11401D-01	0.97050D-01	0.18567D 00	0.59756D-02	0.42854D-02	0.36246D-02
2.0000	0.73412D-02	0.60945D-01	0.13169D 00	0.39875D-02	0.28673D-02	0.24309D-02
2.3333	0.48431D-02	0.38370D-01	0.93328D-01	0.26750D-02	0.19281D-02	0.16361D-02
2.6667	0.32313D-02	0.24225D-01	0.66090D-01	0.18026D-02	0.13022D-02	0.11059D-02
3.0000	0.21704D-02	0.15347D-01	0.46772D-01	0.12159D-02	0.88295D-03	0.75042D-03
3.3333	0.14649D-02	0.97497D-02	0.33083D-01	0.82360D-03	0.60093D-03	0.51110D-03
3.6667	0.[illegible]9295D-03	0.62178D-02	0.23389D-01	0.55487D-03	0.41042D-03	0.34925D-03
4.0000	0.67553D-03	0.39836D-02	0.16529D-01	0.37637D-03	0.28116D-03	0.23946D-03
4.3333	0.46127D-03	0.25537D-02	0.11677D-01	0.25510D-03	0.19322D-03	0.16464D-03
4.6667	0.31596D-03	0.16517D-02	0.82468D-02	0.18241D-03	0.13313D-03	0.11350D-03
5.0000	0.21707D-03	0.10703D-02	0.58226D-02	0.12583D-03	0.91959D-04	0.78436D-04
5.3333	0.14954D-03	0.69749D-03	0.41101D-02	0.87012D-04	0.63662D-04	0.54322D-04
5.6667	0.10327D-03	0.45640D-03	0.29007D-02	0.60297D-04	0.44162D-04	0.37696D-04
6.0000	0.71482D-04	0.30004D-03	0.20468D-02	0.41865D-04	0.30690D-04	0.26203D-04
6.3333	0.49579D-04	0.19818D-03	0.14441D-02	0.29117D-04	0.21362D-04	0.18242D-04
6.6667	0.34451D-04	0.13151D-03	0.10187D-02	0.20283D-04	0.14890D-04	0.12715D-04
7.0000	0.23973D-04	0.87472D-04	0.71851D-03	0.14143D-04	0.10391D-04	0.88694D-05
7.3333	0.16714D-04	0.58708D-04	0.50674D-03	0.98807D-05	0.72564D-05	0.61581D-05
7.6667	0.11665D-04	0.39483D-04	0.35736D-03	0.69076D-05	0.50683D-05	0.43134D-05
8.0000	0.81505D-05	0.26563D-04	0.25199D-03	0.48330D-05	0.35358D-05	0.29975D-05
8.3333	0.57002D-05	0.17976D-04	0.17768D-03	0.33831D-05	0.24565D-05	0.20677D-05
8.6667	0.39901D-05	0.12292D-04	0.12528D-03	0.23677D-05	0.16379D-05	0.14032D-05
9.0000	0.27924D-05	0.83360D-05	0.88324D-04	0.16533D-05	0.11276D-05	0.91806D-06
9.3333	0.19533D-05	0.57482D-05	0.62266D-04	0.11384D-05	0.69283D-06	0.55052D-06
9.6667	0.13617D-05	0.39435D-05	0.43891D-04	0.70573D-06	0.33982D-06	0.25497D-06
10.0000	0.93371D-06	0.27029D-05	0.30930D-04	0.0	0.0	0.0

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

THE TRANSFORM METHOD ALGORITHM RESULTS ARE LISTED BELOW FOR THE FOLLOWING PROBLEM PARAMETERS

SLABWIDTH=10.0 CM.
TOTAL MACROSCOPIC CROSS SECTION=1.0000 1/CM.
SCATTERING MACROSCOPIC CROSS SECTION=0.2000 1/CM.
NUMBER OF SPATIAL MESH INCREMENTS= 30 .
SIZE OF EACH SPATIAL MESH INCREMENT=0.33333 CM.
ANGULAR QUADRATURE SET USED=SIX ORDINATE GAUSS-LEGENDRE.
SOURCE=ZERO.
ACCELERATION PARAMETER ALPHA= 0.500 .

THE FRACTIONAL DIFFERENCE CONVERGENCE BOUND, 0.10000E-03, WAS MET AFTER 10 ITERATIONS.

THE CONVERGED ITERATE ANGULAR FLUXES ARE LISTED

X.CM.	MU(1)	MU(2)	MU(3)	MU(4)	MU(5)	MU(6)
0.0	0.10000D 01	0.10000D 01	0.10000D 01	0.69962D-01	0.45799D-01	0.37758D-01
0.3333	0.26215D 00	0.63221D 00	0.72184D 00	0.36237D-01	0.24892D-01	0.20813D-01
0.6667	0.93793D-01	0.39605D 00	0.51625D 00	0.21644D-01	0.15223D-01	0.12816D-01
1.0000	0.35251D-01	0.24760D 00	0.36780D 00	0.13785D-01	0.97998D-02	0.82759D-02
1.3333	0.15765D-01	0.15485D 00	0.26150D 00	0.90195D-02	0.64465D-02	0.54535D-02
1.6667	0.11401D-01	0.97350D-01	0.18567D 00	0.59756D-02	0.42854D-02	0.36246D-02
2.0000	0.73412D-02	0.60945D-01	0.13169D 00	0.39875D-02	0.28673D-02	0.24309D-02
2.3333	0.48431D-02	0.38370D-01	0.93328D-01	0.26750D-02	0.19281D-02	0.16361D-02
2.6667	0.32313D-02	0.24226D-01	0.66090D-01	0.18026D-02	0.13022D-02	0.11058D-02
3.0000	0.21704D-02	0.15343D-01	0.46772D-01	0.12159D-02	0.86295D-03	0.75042D-03
3.3333	0.14649D-02	0.97497D-02	0.33083D-01	0.82369D-03	0.60093D-03	0.51110D-03
3.6667	0.99295D-03	0.62178D-02	0.23389D-01	0.56487D-03	0.41042D-03	0.34929D-03
4.0000	0.67553D-03	0.39836D-02	0.16529D-01	0.38637D-03	0.28118D-03	0.23946D-03
4.3333	0.46127D-03	0.25587D-02	0.11677D-01	0.26510D-03	0.19322D-03	0.16464D-03
4.6667	0.31596D-03	0.16517D-02	0.82468D-02	0.18241D-03	0.13313D-03	0.11350D-03
5.0000	0.21707D-03	0.10703D-02	0.58226D-02	0.12583D-03	0.91959D-04	0.78436D-04
5.3333	0.14954D-03	0.69749D-03	0.41101D-02	0.87012D-04	0.63662D-04	0.54322D-04
5.6667	0.10327D-03	0.45640D-03	0.29007D-02	0.60297D-04	0.44162D-04	0.37656D-04
6.0000	0.71482D-04	0.30004D-03	0.20468D-02	0.41845D-04	0.30690D-04	0.26203D-04
6.3333	0.49579D-04	0.19818D-03	0.14441D-02	0.29117D-04	0.21362D-04	0.18242D-04
6.6667	0.34451D-04	0.13151D-03	0.10187D-02	0.20283D-04	0.14890D-04	0.12715D-04
7.0000	0.23973D-04	0.87672D-04	0.71851D-03	0.14143D-04	0.10391D-04	0.88694D-05
7.3333	0.16714D-04	0.58709D-04	0.50674D-03	0.98807D-05	0.72564D-05	0.61681D-05
7.6667	0.11665D-04	0.39483D-04	0.35736D-03	0.69076D-05	0.50583D-05	0.43134D-05
8.0000	0.81505D-05	0.26563D-04	0.25199D-03	0.48330D-05	0.35358D-05	0.29975D-05
8.3333	0.57002D-05	0.17976D-04	0.17768D-03	0.33831D-05	0.24565D-05	0.20677D-05
8.6667	0.39891D-05	0.12293D-04	0.12528D-03	0.23677D-05	0.16379D-05	0.14032D-05
9.0000	0.27924D-05	0.83360D-05	0.88324D-04	0.16533D-05	0.11276D-05	0.91806D-06
9.3333	0.19533D-05	0.57482D-05	0.62266D-04	0.11384D-05	0.69880D-06	0.55052D-06
9.6667	0.13617D-05	0.39435D-05	0.43891D-04	0.70573D-06	0.33982D-06	0.25497D-06
10.0000	0.93371D-06	0.27029D-05	0.30930D-04	0.0	0.0	0.0

NOTE THAT D 01 MEANS TEN TO THE POWER ONE.

COMPARISON OF RESULTS FOR THREE DIFFERENT ALGORITHMS ARE MADE FOR MU(1) PART OF SOLUTION

X,CM.	UNMODIFIED R=30 SOLUTION	TRANSFORM METHOD R=30 SOLUTION	UNMODIFIED R=600 SOLUTION	UNMODIFIED DIFFERENCE	TRANSFORM METHOD DIFFERENCE	TRANSFORM METHOD DIFFERENCE
0.0	0.100000 01	0.100000 01	0.100000 01	0.0	0.0	0.0
0.3333	0.241180 00	0.262150 00	0.299100 00	0.579200-01	0.209640-01	0.369560-01
0.6667	0.746250-01	0.837930-01	0.101900 00	0.272720-01	0.916840-02	0.181030-01
1.0000	0.320710-01	0.352510-01	0.419640-01	0.989240-02	0.317990-02	0.671240-02
1.3333	0.176430-01	0.187650-01	0.210210-01	0.337800-02	0.112200-02	0.225600-02
1.6667	0.102250-01	0.114010-01	0.121430-01	0.121790-02	0.475410-03	0.742480-03
2.0000	0.708270-02	0.734120-02	0.759840-02	0.515710-03	0.258450-03	0.257250-03
2.3333	0.467530-02	0.484310-02	0.494520-02	0.269910-03	0.167790-03	0.102120-03
2.6667	0.311340-02	0.323130-02	0.328060-02	0.167190-03	0.117920-03	0.492720-04
3.0000	0.218540-02	0.217040-02	0.219880-02	0.113390-03	0.649420-04	0.283950-04
3.3333	0.140340-02	0.146490-02	0.148320-02	0.798360-04	0.615560-04	0.162700-04
3.6667	0.948390-03	0.982950-03	0.100530-02	0.569360-04	0.445510-04	0.123350-04
4.0000	0.643410-03	0.675580-03	0.684160-03	0.407420-04	0.321700-04	0.857290-05
4.3333	0.438080-03	0.461270-03	0.467240-03	0.291600-04	0.231870-04	0.597330-05
4.6667	0.299660-03	0.315350-03	0.320130-03	0.204720-04	0.166930-04	0.417930-05
5.0000	0.205050-03	0.217070-03	0.219990-03	0.149340-04	0.120090-04	0.292450-05
5.3333	0.140890-03	0.149540-03	0.151580-03	0.106860-04	0.863520-05	0.204790-05
5.6667	0.970580-04	0.103270-03	0.104710-03	0.764930-05	0.621390-05	0.143540-05
6.0000	0.670110-04	0.714820-04	0.724890-04	0.547870-05	0.447130-05	0.100740-05
6.3333	0.463610-04	0.495780-04	0.502870-04	0.392570-05	0.321670-05	0.707990-06
6.6667	0.321330-04	0.344510-04	0.349490-04	0.281550-05	0.231810-05	0.498420-06
7.0000	0.223080-04	0.239780-04	0.243300-04	0.202130-05	0.167030-05	0.351500-06
7.3333	0.155000-04	0.167140-04	0.169520-04	0.145250-05	0.120410-05	0.248340-06
7.6667	0.107970-04	0.116550-04	0.118410-04	0.104420-05	0.906410-06	0.175750-06
8.0000	0.752400-05	0.815050-05	0.827510-05	0.751060-06	0.628490-06	0.124570-06
8.3333	0.524600-05	0.569820-05	0.578650-05	0.540460-06	0.452070-06	0.883370-07
8.6667	0.366300-05	0.393910-05	0.400180-05	0.338760-06	0.326230-06	0.627270-07
9.0000	0.255700-05	0.279240-05	0.283680-05	0.279770-06	0.235360-06	0.444090-07
9.3333	0.178370-05	0.195330-05	0.198440-05	0.200790-06	0.169580-06	0.311100-07
9.6667	0.123370-05	0.136170-05	0.138300-05	0.143210-06	0.122000-06	0.212040-07
10.0000	0.848000-06	0.933710-06	0.949590-06	0.101590-06	0.857130-07	0.158790-07

UNMODIFIED DIFFERENCE=(UNMODIFIED R=600 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
OBSERVE THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION VALUES.
TRANSFORM METHOD DIFFERENCE IN NEXT TO LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
A POSITIVE QUANTITY HERE MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.
TRANSFORM METHOD DIFFERENCE IN LAST COLUMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30 SOLUTION).

THE KEY TO INTERPRETING THE TABLE IS:
1. THE FINER SPATIAL MESH, R=600, PRODUCES LARGER SOLUTION VALUES THAN THE R=30 MESH.
2. A POSITIVE QUANTITY IN COLUMNS 7 AND 8 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZATION ERROR.
3. A NEGATIVE QUANTITY IN COLUMN 7 MEANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.
4. A POSITIVE QUANTITY IN COLUMN 7 AND NEGATIVE IN COLUMN 8 MEANS THE TRANSFORM PRODUCED A SOLUTION IN THE DIRECTION OF IMPROVED DISCRETIZATION ERROR BUT OVERSHOT THE R=600 SOLUTION.

THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(1) IS 0.101030 00 .
THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR MU(1) IS 0.652670-01 .

THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION IS 0.173520 00 .
THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 AND TRANSFORM METHOD SOLUTIONS IS 0.813130-01 .

COMPARISON OF RESULTS FOR THREE DIFFERENT ALGORITHMS ARE MADE FOR MU(2) PART OF SOLUTION

X.CM.	UNMODIFIED R=30 SOLUTION	TRANSFORM METHOD R=30 SOLUTION	UNMODIFIED R=600 SOLUTION	UNMODIFIED DIFFERENCE	TRANSFORM METHOD DIFFERENCE	TRANSFORM METHOD DIFFERENCE
0.0	0.100000D 01	0.100000D 01	0.100000D 01	0.0	0.0	0.0
0.3333	0.608520D 00	0.612210D 00	0.612740D 00	0.421560D-02	0.368790D-02	0.527810D-03
0.6667	0.391030D 00	0.396050D 00	0.397490D 00	0.645820D-02	0.502510D-02	0.144310D-02
1.0000	0.242800D 00	0.247600D 00	0.249230D 00	0.642710D-02	0.480170D-02	0.162540D-02
1.3333	0.150890D 00	0.154890D 00	0.156320D 00	0.542920D-02	0.399590D-02	0.143240D-02
1.6667	0.939510D-01	0.970500D-01	0.981850D-01	0.423360D-02	0.309370D-02	0.113490D-02
2.0000	0.586410D-01	0.609450D-01	0.617950D-01	0.315450D-02	0.230420D-02	0.850320D-03
2.3333	0.317030D-01	0.333700D-01	0.339860D-01	0.228280D-02	0.166650D-02	0.616030D-03
2.6667	0.230430D-01	0.242260D-01	0.246620D-01	0.161920D-02	0.118270D-02	0.436460D-03
3.0000	0.144150D-01	0.152430D-01	0.155470D-01	0.113190D-02	0.827530D-03	0.304370D-03
3.3333	0.917670D-02	0.974970D-02	0.995940D-02	0.782760D-03	0.573010D-03	0.209740D-03
3.6667	0.582410D-02	0.621740D-02	0.636100D-02	0.536910D-03	0.393690D-03	0.143220D-03
4.0000	0.371170D-02	0.398060D-02	0.407770D-02	0.366010D-03	0.268910D-03	0.970990D-04
4.3333	0.237570D-02	0.255870D-02	0.262410D-02	0.248370D-03	0.182910D-03	0.654570D-04
4.6667	0.152760D-02	0.165170D-02	0.169560D-02	0.167970D-03	0.124040D-03	0.439300D-04
5.0000	0.984920D-03	0.107030D-02	0.110030D-02	0.113350D-03	0.839660D-04	0.293810D-04
5.3333	0.641710D-03	0.697490D-03	0.717090D-03	0.753820D-04	0.567630D-04	0.195990D-04
5.6667	0.418000D-03	0.456400D-03	0.469440D-03	0.514430D-04	0.383940D-04	0.130500D-04
6.0000	0.274070D-03	0.300040D-03	0.308720D-03	0.346520D-04	0.259740D-04	0.867820D-05
6.3333	0.180590D-03	0.198130D-03	0.203950D-03	0.233590D-04	0.175910D-04	0.576770D-05
6.6667	0.119580D-03	0.131510D-03	0.135350D-03	0.157660D-04	0.119530D-04	0.383310D-05
7.0000	0.795600D-04	0.876720D-04	0.902210D-04	0.106610D-04	0.811200D-05	0.254860D-05
7.3333	0.531800D-04	0.587080D-04	0.604040D-04	0.722360D-05	0.552760D-05	0.169600D-05
7.6667	0.357060D-04	0.394830D-04	0.406130D-04	0.490730D-05	0.377710D-05	0.113020D-05
8.0000	0.240740D-04	0.266630D-04	0.274170D-04	0.334290D-05	0.258840D-05	0.754480D-06
8.3333	0.162960D-04	0.180750D-04	0.185800D-04	0.228400D-05	0.177930D-05	0.504700D-06
8.6667	0.110720D-04	0.122490D-04	0.126370D-04	0.156530D-05	0.122690D-05	0.338360D-06
9.0000	0.754730D-05	0.839600D-05	0.862340D-05	0.107610D-05	0.848670D-06	0.227390D-06
9.3333	0.515950D-05	0.574320D-05	0.590130D-05	0.741930D-06	0.588730D-06	0.153040D-06
9.6667	0.353400D-05	0.394350D-05	0.404650D-05	0.512490D-06	0.409500D-06	0.102990D-06
10.0000	0.241840D-05	0.270290D-05	0.277320D-05	0.354740D-06	0.284460D-06	0.702810D-07

UNMODIFIED DIFFERENCE=(UNMODIFIED R=600 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
OBSERVE THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION VALUES.
TRANSFORM METHOD DIFFERENCE IN NEXT TO LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
A POSITIVE QUANTITY HERE MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.
TRANSFORM METHOD DIFFERENCE IN LAST COLUMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30 SOLUTION).

THE KEY TO INTERPRETING THE TABLE IS:

1. THE FINER SPATIAL MESH, R=600, PRODUCES LARGER SOLUTION VALUES THAN THE R=30 MESH.
2. A POSITIVE QUANTITY IN COLUMNS 7 AND 8 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZATION ERROR.
3. A NEGATIVE QUANTITY IN COLUMN 7 MEANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.
4. A POSITIVE QUANTITY IN COLUMN 7 AND NEGATIVE IN COLUMN 8 MEANS THE TRANSFORM PRODUCED A SOLUTION IN THE DIRECTION OF IMPROVED DISCRETIZATION ERROR BUT OVERSHOT THE R=600 SOLUTION.

THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(2) IS 0.374110D-01 .
THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR MU(2) IS 0.901810D-02 .

THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION IS 0.173520D 00 .
THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 AND TRANSFORM METHOD SOLUTIONS IS 0.813130D-01 .

COMPARISON OF RESULTS FOR THREE DIFFERENT ALGORITHMS ARE MADE FOR MU(3) PART OF SOLUTION

X,CM.	UNMODIFIED R=30 SOLUTION	TRANSFORM METHOD R=30 SOLUTION	UNMODIFIED R=600 SOLUTION	UNMODIFIED DIFFERENCE	TRANSFORM METHOD DIFFERENCE	TRANSFORM METHOD DIFFERENCE
0.0	0.10000D 01	0.10000D 01	0.10000D 01	0.0	0.0	0.0
0.3333	0.72020D 00	0.72184D 00	0.72142D 00	0.12220D-02	0.16379D-02	-0.41593D-03
0.6667	0.51351D 00	0.51625D 00	0.51629D 00	0.27743D-02	0.27403D-02	0.33997D-04
1.0000	0.36472D 00	0.36730D 00	0.36809D 00	0.33605D-02	0.30309D-02	0.27951D-03
1.3333	0.25852D 00	0.26150D 00	0.26185D 00	0.33415D-02	0.29845D-02	0.35668D-03
1.6667	0.18299D 00	0.18567D 00	0.18603D 00	0.30355D-02	0.26817D-02	0.35383D-03
2.0000	0.12939D 00	0.13169D 00	0.13201D 00	0.26207D-02	0.23018D-02	0.31898D-03
2.3333	0.91412D-01	0.93327D-01	0.93601D-01	0.21891D-02	0.19153D-02	0.27374D-03
2.6667	0.64532D-01	0.66090D-01	0.66318D-01	0.17853D-02	0.15580D-02	0.22800D-03
3.0000	0.45127D-01	0.46772D-01	0.46958D-01	0.14310D-02	0.12456D-02	0.18547D-03
3.3333	0.32101D-01	0.33083D-01	0.33231D-01	0.11306D-02	0.98212D-03	0.14851D-03
3.6667	0.22623D-01	0.23339D-01	0.23507D-01	0.88311D-03	0.76577D-03	0.11733D-03
4.0000	0.15938D-01	0.16529D-01	0.16621D-01	0.68326D-03	0.59157D-03	0.91687D-04
4.3333	0.11224D-01	0.11677D-01	0.11749D-01	0.52442D-03	0.45344D-03	0.70939D-04
4.6667	0.79015D-02	0.82468D-02	0.83013D-02	0.39978D-03	0.34525D-03	0.54535D-04
5.0000	0.56313D-02	0.59226D-02	0.59643D-02	0.33293D-03	0.26137D-03	0.41614D-04
5.3333	0.39132D-02	0.41101D-02	0.41417D-02	0.22845D-03	0.19658D-03	0.31549D-04
5.6667	0.27530D-02	0.29007D-02	0.29245D-02	0.17149D-03	0.14766D-03	0.23828D-04
6.0000	0.19395D-02	0.20468D-02	0.20647D-02	0.12522D-03	0.11031D-03	0.17904D-04
6.3333	0.13619D-02	0.14441D-02	0.14575D-02	0.95530D-04	0.82131D-04	0.13399D-04
6.6667	0.95771D-03	0.10147D-02	0.10287D-02	0.70953D-04	0.60960D-04	0.99930D-05
7.0000	0.67339D-03	0.71851D-03	0.72594D-03	0.52551D-04	0.45122D-04	0.74291D-05
7.3333	0.47343D-03	0.50674D-03	0.51225D-03	0.38872D-04	0.33315D-04	0.55073D-05
7.6667	0.33282D-03	0.35736D-03	0.36143D-03	0.28614D-04	0.24547D-04	0.40722D-05
8.0000	0.23355D-03	0.25199D-03	0.25500D-03	0.21045D-04	0.18642D-04	0.30041D-05
8.3333	0.16445D-03	0.17768D-03	0.17989D-03	0.15450D-04	0.13233D-04	0.22115D-05
8.6667	0.11558D-03	0.12528D-03	0.12690D-03	0.11322D-04	0.96970D-05	0.16248D-05
9.0000	0.86232D-04	0.88324D-04	0.89515D-04	0.32832D-05	0.70916D-05	0.11916D-05
9.3333	0.57087D-04	0.62266D-04	0.63138D-04	0.60508D-05	0.51785D-05	0.87229D-06
9.6667	0.40115D-04	0.43891D-04	0.44528D-04	0.44136D-05	0.37762D-05	0.63737D-06
10.0000	0.28181D-04	0.30930D-04	0.31396D-04	0.32149D-05	0.27492D-05	0.46570D-06

UNMODIFIED DIFFERENCE=(UNMODIFIED R=600 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
OBSERVE THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION VALUES.
TRANSFORM METHOD DIFFERENCE IN NEXT TO LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
A POSITIVE QUANTITY HERE MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.
TRANSFORM METHOD DIFFERENCE IN LAST COLUMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30 SOLUTION).

THE KEY TO INTERPRETING THE TABLE IS:
1. THE FINER SPATIAL MESH, R=600, PRODUCES LARGER SOLUTION VALUES THAN THE R=30 MESH.
2. A POSITIVE QUANTITY IN COLUMNS 7 AND 8 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZATION ERROR.
3. A NEGATIVE QUANTITY IN COLUMN 7 MEANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.
4. A POSITIVE QUANTITY IN COLUMN 7 AND NEGATIVE IN COLUMN 8 MEANS THE TRANSFORM PRODUCED A SOLUTION IN THE DIRECTION OF IMPROVED DISCRETIZATION ERROR BUT OVERSHOT THE R=600 SOLUTION.

THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(3) IS 0.26569D-01 .
THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR MU(3) IS 0.30947D-02 .

THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION IS 0.17352D 00 .
THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 AND TRANSFORM METHOD SOLUTIONS IS 0.81313D-01 .

COMPARISON OF RESULTS FOR THREE DIFFERENT ALGORITHMS ARE MADE FOR MU(4) PART OF SOLUTION

X.CM.	UNMODIFIED R=30 SOLUTION	TRANSFORM METHOD R=30 SOLUTION	UNMODIFIED R=600 SOLUTION	UNMODIFIED DIFFERENCE	TRANSFORM METHOD DIFFERENCE	TRANSFORM METHOD DIFFERENCE
0.0	0.699620D-01	0.699620D-01	0.699570D-01	-0.732610D-13	0.423830D-13	-0.115540D-12
0.3333	0.355740D-01	0.362370D-01	0.371810D-01	0.160770D-02	0.663740D-03	0.944010D-03
0.6667	0.212230D-01	0.216440D-01	0.221450D-01	0.922390D-03	0.421510D-03	0.500850D-03
1.0000	0.135170D-01	0.137350D-01	0.140270D-01	0.490520D-03	0.266550D-03	0.221860D-03
1.3333	0.882340D-02	0.901980D-02	0.912450D-02	0.235040D-03	0.191430D-03	0.104650D-03
1.6667	0.583160D-02	0.597550D-02	0.603320D-02	0.201630D-03	0.144070D-03	0.575550D-04
2.0000	0.387910D-02	0.396750D-02	0.402390D-02	0.145600D-03	0.109420D-03	0.363800D-04
2.3333	0.259250D-02	0.267500D-02	0.269930D-02	0.107350D-03	0.824860D-04	0.246760D-04
2.6667	0.174110D-02	0.180250D-02	0.182020D-02	0.790950D-04	0.615380D-04	0.175570D-04
3.0000	0.117430D-02	0.121990D-02	0.123230D-02	0.579910D-04	0.454920D-04	0.124590D-04
3.3333	0.795200D-03	0.828600D-03	0.837490D-03	0.422910D-04	0.333950D-04	0.289620D-05
3.6667	0.540480D-03	0.564870D-03	0.571190D-03	0.307050D-04	0.243850D-04	0.631540D-05
4.0000	0.368530D-03	0.386370D-03	0.390850D-03	0.222230D-04	0.177490D-04	0.447130D-05
4.3333	0.252210D-03	0.265100D-03	0.268260D-03	0.160440D-04	0.128850D-04	0.315940D-05
4.6667	0.173070D-03	0.182410D-03	0.184630D-03	0.115580D-04	0.933840D-05	0.222570D-05
5.0000	0.119070D-03	0.125330D-03	0.127410D-03	0.833430D-05	0.676140D-05	0.157300D-05
5.3333	0.821190D-04	0.870120D-04	0.881220D-04	0.600250D-05	0.489270D-05	0.110990D-05
5.6667	0.567570D-04	0.602970D-04	0.610800D-04	0.432310D-05	0.353940D-05	0.783620D-06
6.0000	0.393040D-04	0.418650D-04	0.424180D-04	0.311410D-05	0.256030D-05	0.553640D-06
6.3333	0.272650D-04	0.291170D-04	0.295090D-04	0.224390D-05	0.185200D-05	0.391920D-06
6.6667	0.189430D-04	0.202830D-04	0.205600D-04	0.161750D-05	0.133980D-05	0.277720D-06
7.0000	0.131790D-04	0.141480D-04	0.143450D-04	0.116640D-05	0.969340D-06	0.197070D-06
7.3333	0.917930D-05	0.988070D-05	0.100210D-04	0.841380D-06	0.701360D-06	0.140010D-06
7.6667	0.640010D-05	0.690760D-05	0.700710D-05	0.607040D-06	0.507510D-06	0.995330D-07
8.0000	0.446570D-05	0.483330D-05	0.490360D-05	0.437930D-06	0.367220D-06	0.705760D-07
8.3333	0.311740D-05	0.338310D-05	0.343280D-05	0.315460D-06	0.265710D-06	[illegible]
8.6667	0.217540D-05	0.236770D-05	0.240130D-05	0.225910D-06	0.192370D-06	0.335410D-07
9.0000	0.151340D-05	0.165330D-05	0.167110D-05	0.157740D-06	0.139920D-06	0.178220D-07
9.3333	0.105460D-05	0.113940D-05	0.113350D-05	0.959360D-07	0.103860D-06	-0.492430D-08
9.6667	0.627990D-06	0.705730D-06	0.664940D-06	0.370390D-07	0.778320D-07	-0.407930D-07
10.0000	0.0	0.0	0.0	0.0	0.0	0.0 .

UNMODIFIED DIFFERENCE=(UNMODIFIED R=600 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
OBSERVE THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION VALUES.
TRANSFORM METHOD DIFFERENCE IN NEXT TO LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
A POSITIVE QUANTITY HERE MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.
TRANSFORM METHOD DIFFERENCE IN LAST COLUMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30 SOLUTION).

THE KEY TO INTERPRETING THE TABLE IS:
1. THE FINER SPATIAL MESH, R=600, PRODUCES LARGER SOLUTION VALUES THAN THE R=30 MESH.
2. A POSITIVE QUANTITY IN COLUMNS 7 AND 8 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZATION ERROR.
3. A NEGATIVE QUANTITY IN COLUMN 7 MEANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.
4. A POSITIVE QUANTITY IN COLUMN 7 AND NEGATIVE IN COLUMN 8 MEANS THE TRANSFORM PRODUCED A SOLUTION IN THE DIRECTION OF IMPROVED DISCRETIZATION ERROR BUT OVERSHOT THE R=600 SOLUTION.

THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(4) IS 0.406090-02 .
THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR MU(4) IS 0.19507D-02 .

THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION IS 0.17352D 00 .
THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 AND TRANSFORM METHOD SOLUTIONS IS 0.81313D-01 .

COMPARISON OF RESULTS FOR THREE DIFFERENT ALGORITHMS ARE MADE FOR MU(5) PART OF SOLUTION

X,CM.	UNMODIFIED R=30 SOLUTION	TRANSFORM METHOD R=30 SOLUTION	UNMODIFIED R=600 SOLUTION	UNMODIFIED DIFFERENCE	TRANSFORM METHOD DIFFERENCE	TRANSFORM METHOD DIFFERENCE
0.0	0.457930D-01	0.457930D-01	0.457930D-01	-0.241520D-12	0.140150D-12	-0.351870D-12
0.3333	0.245090D-01	0.248920D-01	0.254060D-01	0.897850D-03	0.383110D-03	0.514740D-03
0.6667	0.149630D-01	0.152230D-01	0.155030D-01	0.540080D-03	0.260090D-03	0.279920D-03
1.0000	0.962200D-02	0.979980D-02	0.992880D-02	0.306770D-03	0.177790D-03	0.128980D-03
1.3333	0.631470D-02	0.644650D-02	0.651110D-02	0.196380D-03	0.131620D-03	0.646220D-04
1.6667	0.418440D-02	0.428540D-02	0.432300D-02	0.138600D-03	0.101010D-03	0.375930D-04
2.0000	0.279000D-02	0.286730D-02	0.289200D-02	0.102010D-03	0.773420D-04	0.246670D-04
2.3333	0.186950D-02	0.192810D-02	0.194530D-02	0.757650D-04	0.585630D-04	0.172220D-04
2.6667	0.125930D-02	0.130320D-02	0.131440D-02	0.550870D-04	0.439280D-04	0.112590D-04
3.0000	0.860460D-03	0.892950D-03	0.901720D-03	0.412610D-04	0.324860D-04	0.877310D-05
3.3333	0.577020D-03	0.600930D-03	0.607190D-03	0.301760D-04	0.239160D-04	0.626590D-05
3.6667	0.392890D-03	0.410400D-03	0.414860D-03	0.219660D-04	0.175050D-04	0.446050D-05
4.0000	0.268410D-03	0.281130D-03	0.284350D-03	0.159350D-04	0.127690D-04	0.316520D-05
4.3333	0.183930D-03	0.193220D-03	0.195460D-03	0.115340D-04	0.928080D-05	0.224270D-05
4.6667	0.126380D-03	0.133130D-03	0.134720D-03	0.833500D-05	0.674830D-05	0.158670D-05
5.0000	0.870630D-04	0.919590D-04	0.930810D-04	0.601810D-05	0.489610D-05	0.112210D-05
5.3333	0.601120D-04	0.636620D-04	0.644560D-04	0.434330D-05	0.354970D-05	0.793960D-06
5.6667	0.415890D-04	0.441620D-04	0.447230D-04	0.313410D-05	0.257260D-05	0.561520D-06
6.0000	0.288260D-04	0.306910D-04	0.310870D-04	0.226160D-05	0.186400D-05	0.397630D-06
6.3333	0.200120D-04	0.213620D-04	0.216440D-04	0.163210D-05	0.135040D-05	0.281710D-06
6.6667	0.139120D-04	0.148900D-04	0.150900D-04	0.117800D-05	0.978340D-06	0.199640D-06
7.0000	0.968180D-05	0.103910D-04	0.105320D-04	0.850110D-06	0.706770D-06	0.141350D-06
7.3333	0.674300D-05	0.725640D-05	0.735620D-05	0.613190D-06	0.513440D-06	0.997400D-07
7.6667	0.469640D-05	0.506830D-05	0.513810D-05	0.441650D-06	0.371840D-06	0.698100D-07
8.0000	0.324670D-05	0.351580D-05	0.356380D-05	0.317040D-06	0.269030D-06	0.480100D-07
8.3333	0.226240D-05	0.245660D-05	0.248830D-05	0.225930D-06	0.194090D-06	0.313410D-07
8.6667	0.154910D-05	0.168790D-05	0.170750D-05	0.158470D-06	0.138870D-06	0.195960D-07
9.0000	0.103060D-05	0.112760D-05	0.113790D-05	0.107320D-06	0.970540D-07	0.102710D-07
9.3333	0.635520D-06	0.698800D-06	0.702450D-06	0.669350D-07	0.632850D-07	0.364740D-08
9.6667	0.307390D-06	0.339820D-06	0.340230D-06	0.328430D-07	0.324360D-07	0.406710D-09
10.0000	0.0	0.0	0.0	0.0	0.0	0.0

UNMODIFIED DIFFERENCE=(UNMODIFIED R=600 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
OBSERVE THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION VALUES.
TRANSFORM METHOD DIFFERENCE IN NEXT TO LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
A POSITIVE QUANTITY HERE MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.
TRANSFORM METHOD DIFFERENCE IN LAST COLUMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30 SOLUTION).

THE KEY TO INTERPRETING THE TABLE IS:
1. THE FINER SPATIAL MESH, R=600, PRODUCES LARGER SOLUTION VALUES THAN THE R=30 MESH.
2. A POSITIVE QUANTITY IN COLUMNS 7 AND 8 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZATION ERROR.
3. A NEGATIVE QUANTITY IN COLUMN 7 MEANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.
4. A POSITIVE QUANTITY IN COLUMN 7 AND NEGATIVE IN COLUMN 8 MEANS THE TRANSFORM PRODUCED A SOLUTION IN THE DIRECTION OF IMPROVED DISCRETIZATION ERROR BUT OVERSHOT THE R=600 SOLUTION.

THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(5) IS 0.246410-02 .
THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR MU(5) IS 0.110950-02 .

THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION IS 0.173520 00 .
THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 AND TRANSFORM METHOD SOLUTIONS IS 0.813130-01 .

COMPARISON OF RESULTS FOR THREE DIFFERENT ALGORITHMS ARE MADE FOR MU(6) PART OF SOLUTION

X.CM.	UNMODIFIED R=30 SOLUTION	TRANSFORM METHOD R=30 SOLUTION	UNMODIFIED R=600 SOLUTION	UNMODIFIED DIFFERENCE	TRANSFORM METHOD DIFFERENCE	TRANSFORM METHOD DIFFERENCE
0.0	0.377590D-01	0.377590D-01	0.377590D-01	-0.235310D-11	0.136190D-11	-0.371500D-11
0.3333	0.205100D-01	0.208130D-01	0.212120D-01	0.702270D-03	0.302180D-03	0.399290D-03
0.6667	0.126040D-01	0.129160D-01	0.130340D-01	0.429480D-03	0.211230D-03	0.218250D-03
1.0000	0.812890D-02	0.827590D-02	0.837920D-02	0.249230D-03	0.146950D-03	0.102290D-03
1.3333	0.534330D-02	0.545350D-02	0.559570D-02	0.162400D-03	0.110200D-03	0.522030D-04
1.6667	0.354470D-02	0.362960D-02	0.366050D-02	0.115810D-03	0.846380D-04	0.309230D-04
2.0000	0.236570D-02	0.243030D-02	0.245140D-02	0.856730D-04	0.651590D-04	0.205150D-04
2.3333	0.158460D-02	0.163610D-02	0.165040D-02	0.637970D-04	0.494120D-04	0.143850D-04
2.6667	0.106940D-02	0.110580D-02	0.111610D-02	0.473030D-04	0.370220D-04	0.102520D-04
3.0000	0.722950D-03	0.750420D-03	0.757790D-03	0.349410D-04	0.274710D-04	0.737010D-05
3.3333	0.489860D-03	0.511120D-03	0.516370D-03	0.265070D-04	0.202370D-04	0.527020D-05
3.6667	0.334440D-03	0.349290D-03	0.353040D-03	0.185350D-04	0.148300D-04	0.375500D-05
4.0000	0.228630D-03	0.239450D-03	0.242130D-03	0.134960D-04	0.108290D-04	0.268830D-05
4.3333	0.156760D-03	0.164640D-03	0.166530D-03	0.977670D-05	0.788490D-05	0.189180D-05
4.6667	0.107770D-03	0.113500D-03	0.114840D-03	0.707140D-05	0.573180D-05	0.133960D-05
5.0000	0.742740D-04	0.784350D-04	0.793840D-04	0.510990D-05	0.416180D-05	0.946090D-06
5.3333	0.513030D-04	0.543220D-04	0.549930D-04	0.369050D-05	0.301960D-05	0.673590D-06
5.6667	0.355060D-04	0.376960D-04	0.381710D-04	0.266450D-05	0.218980D-05	0.475020D-06
6.0000	0.246160D-04	0.262030D-04	0.265400D-04	0.192400D-05	0.159760D-05	0.336410D-06
6.3333	0.170910D-04	0.182420D-04	0.184800D-04	0.133900D-05	0.115070D-05	0.238230D-06
6.6667	0.118810D-04	0.127150D-04	0.128830D-04	0.100260D-05	0.833930D-06	0.168530D-06
7.0000	0.826520D-05	0.886940D-05	0.898850D-05	0.723330D-06	0.604160D-06	0.119130D-06
7.3333	0.575070D-05	0.619910D-05	0.627200D-05	0.521230D-06	0.437410D-06	0.832910D-07
7.6667	0.399730D-05	0.431340D-05	0.437190D-05	0.374650D-06	0.316220D-06	0.584300D-07
8.0000	0.276960D-05	0.299750D-05	0.303750D-05	0.267950D-06	0.227390D-06	0.400600D-07
8.3333	0.190460D-05	0.206770D-05	0.209440D-05	0.189760D-06	0.163100D-06	0.266590D-07
8.6667	0.129920D-05	0.140320D-05	0.142100D-05	0.131750D-06	0.114950D-06	0.168290D-07
9.0000	0.839840D-06	0.918060D-06	0.927750D-06	0.879100D-07	0.782240D-07	0.968620D-08
9.3333	0.501630D-06	0.550520D-06	0.555310D-06	0.536340D-07	0.488370D-07	0.479590D-08
9.6667	0.231530D-06	0.254970D-06	0.257020D-06	0.254910D-07	0.234410D-07	0.204520D-08
10.0000	0.0	0.0	0.0	0.0	0.0	0.0

UNMODIFIED DIFFERENCE=(UNMODIFIED R=600 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
OBSERVE THAT THIS QUANTITY IS POSITIVE FOR ALL SOLUTION VALUES.
TRANSFORM METHOD DIFFERENCE IN NEXT TO LAST COLUMN=(TRANSFORM METHOD R=30 SOLUTION)-(UNMODIFIED R=30 SOLUTION).
A POSITIVE QUANTITY HERE MEANS THE TRANSFORM RESULT IS IN THE PROPER DIRECTION.
TRANSFORM METHOD DIFFERENCE IN LAST COLUMN=(UNMODIFIED R=600 SOLUTION)-(TRANSFORM METHOD R=30 SOLUTION).

THE KEY TO INTERPRETING THE TABLE IS:
1. THE FINER SPATIAL MESH, R=600, PRODUCES LARGER SOLUTION VALUES THAN THE R=30 MESH.
2. A POSITIVE QUANTITY IN COLUMNS 7 AND 9 MEANS THE TRANSFORM METHOD DECREASED THE DISCRETIZATION ERROR.
3. A NEGATIVE QUANTITY IN COLUMN 7 MEANS THE TRANSFORM INCREASED THE DISCRETIZATION ERROR.
4. A POSITIVE QUANTITY IN COLUMN 7 AND NEGATIVE IN COLUMN 9 MEANS THE TRANSFORM PRODUCED A SOLUTION IN THE DIRECTION OF IMPROVED DISCRETIZATION ERROR BUT OVERSHOT THE R=600 SOLUTION.

THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION FOR MU(6) IS 0.198340D-02 .
THE BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND TRANSFORM METHOD SOLUTION FOR MU(6) IS 0.873390D-03 .

THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 SOLUTION AND R=30 SOLUTION IS 0.173520 CC .
THE TOTAL BULK SUM OF ABSOLUTE DIFFERENCES IN R=600 AND TRANSFORM METHOD SOLUTIONS IS 0.813130D-01 .

STOP 0
EXECUTION TERMINATED

Thesis
A87 131794
Atkinson
Convergence acceleration and error analysis of the discrete ordinates algorithm in plane geometry.